ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Development of an Autonomous Component Testing System with Reliability Improvement Using Computer Vision and Machine Learning

Hoang Anh Phan[1], Van Tan Duong[2], Mai Nguyen Thi[3], Anh Nguyen Thi[4], Hang Khuat Thi Thu[5], Thang Luu Duc[6], Van Hieu Dang[7], Huu Quoc Dong Tran[8], Thi Thanh Van Nguyen[9] and Thanh Tung Bui[10]

ABSTRACT

This study evaluated computer vision-based models, including Histogram Analysis, Logistic Regression, Sift-SVM, and Deep learning models, in an autonomous testing system developed for smartphone camera modules. System performance was assessed in a practical factory setting with workers operating the system, and metrics such as processing time, sensitivity, specificity, accuracy, and defect rate were evaluated. Based on the results, the Sift-SVM model demonstrated the greatest potential for enhancing the reliability of the system with a processing time of 0.01578 seconds, a sensitivity of 99.811%, and a reduction in the failure rate to 1888 PPM. The study findings suggest that Sift-SVM has the potential to be practically applied in the industry, thus improving the speed and accuracy of automatic defect detection in manufacturing and reducing the defect rate.

Article information:

Keywords: Defect Rate, Pick-and-place Robot, Computer Vision, Machine learning

Article history:
Received: August 18, 2023
Revised: November 23, 2023
Accepted: January 18, 2024
Published: February 10, 2024 (Online)

DOI: 10.37936/ecti-cit.2024181.253854

1. INTRODUCTION

Automatic defect detection technology has become a crucial area of research in modern manufacturing, providing a distinct advantage over traditional manual detection methods [1]. The technology enables adaptation to unknown environments and operates with high precision and efficiency for extended periods. Automatic defect detection can significantly reduce production costs, improve production efficiency and product quality, and provide a robust platform for smart manufacturing.

Computer vision-based detection is one of the most widely used methods for automatic defect detection [2], [3], [4], [5], [6], [7]. This technique involves image acquisition, defect detection, and classification. Computer vision is a popular method because of its speed, accuracy, and low cost. The quality of image acquisition is critical to the difficulty of image processing, and the quality of the processing algorithm directly impacts the model's accuracy and defect detection rate. Defect detection techniques, particularly computer vision and deep learning methods, have gained widespread popularity in recent years. They are essential for automating error detection due to their adaptability and independence from human intervention. State-of-the-art methods in computer vision-based detection include convolutional neural networks(CNNs) [8], [9], [10], [11], You Only Look Once (YOLO) [12], [13], [14], support vector machines (SVMs) [15], [16], [17], and deep belief networks (DBNs) [18], [19]. However, one of the significant challenges in this area is the lack of training data, which directly impacts the model's accuracy [20]. Researchers have proposed transfer learning, data augmentation, and deep-generative models as potential solutions to overcome this challenge.

Numerous studies have been conducted in the field of automatic defect detection using computer vision-based approaches, which has been the subject of extensive research due to its potential to improve production efficiency and quality control in various industries [21], [22], [23], [24], [25], [26], [27], [28], [29], [30]. Jing Yang et al. [31] conducted a study that examines state-of-the-art deep learning methods in defect detection. The study classified defects in various products, such as electronic components, pipes,

[1,2,3,4,5,6,7,8,9,10] The authors are with the VNU University of Engineering and Technology, Vietnam, Email: anh.ph@vnu.edu.vn, duongtanrb@gmail.com, nguyenmaicg03@gmail.com, 147anhnt@gmail.com, thuhangkt01@gmail.com, thangluu3333@gmail.com, hieuvandang3210@gmail.com, dongtran.robotics@gmail.com, vanntt@vnu.edu.vn and tungbt@vnu.edu.vn
[10] Corresponding author: tungbt@vnu.edu.vn

welded parts, and textile materials, into categories and reviewed the main techniques and deep learning methods for defects, highlighting their characteristics, strengths, and shortcomings. Tian Wang et al. [32] proposed a deep convolutional neural network (CNN) for automated quality visual inspection to control product quality for increased efficiency. CNN can effectively extract powerful features from images for defect detection with less prior knowledge and high noise robustness, resulting in fast and accurate detection compared to state-of-the-art methods. Yatao Yang et al. [33] have developed an optimized Visual Geometry Group (VGG) model for efficient laser welding defect classification in battery manufacturing using transfer learning and a pretraining approach. Their proposed model achieved a testing accuracy of 99.87% and demonstrated benefits such as small model size, lower fault positive rate, shorter training time, and prediction time, which makes it suitable for industrial quality inspection.

However, in industry, companies tend to keep their technological and production know-how proprietary to gain a competitive advantage over other companies in the same field. Consequently, there is a lack of publications with the information necessary to address the challenges within factories. In our study, we solve the practical issue that a camera module manufacturing company is facing. In the previous work, we proposed an effective vision system to improve the reliability of pick-and-place robots for the autonomous testing of camera modules [34]. The system confirmed the presence of the camera modules in feeding trays and the placement accuracy of the modules in test sockets by using a simple image processing algorithm based on histogram information. The results of the test of 2000 camera modules showed an accuracy of more than 99.92%. The proposed vision system was a simple and effective solution for pick-and-place systems in the industry. However, these related works did not consider the significance of Six Sigma in manufacturing. Six Sigma is a methodology to improve quality and reduce defects in production processes, which is critical for mass production [35], [36]. Despite the improvement in the accuracy of the model, it has yet to be evaluated in a real-factory environment where conditions and requirements can differ, potentially impacting its effectiveness in reducing the defect rate.

In this study, we focus on testing and evaluating commonly used vision models and applying them to our camera module testing system to determine the most efficient method for detecting improper placement of camera modules, with a focus on reducing the defect rate and thereby achieving a higher Sigma standard. The models included Histogram analysis, Logistic regression, SIFT-SVM, CNN, and YOLOv8 Classify, with the aim of improving the pick-and-place robot's reliability and reducing the system's defect

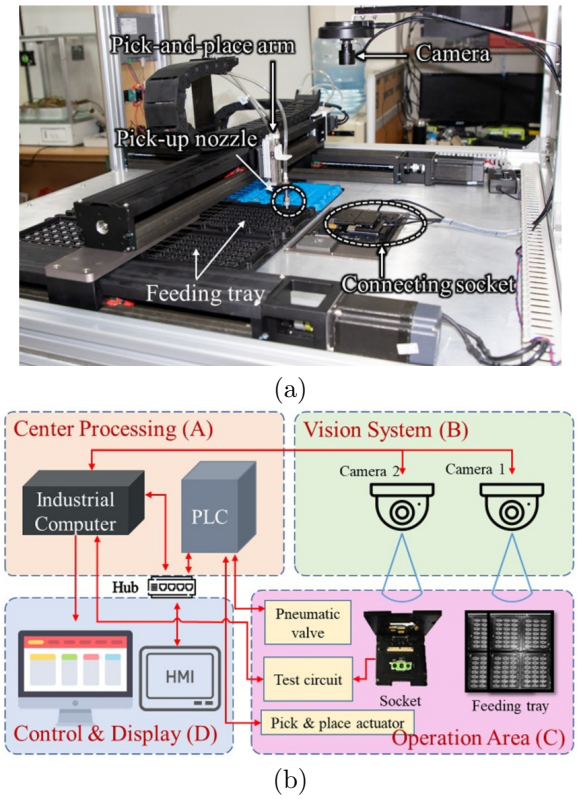


Fig.1: *Autonomous camera module testing system. (a) Actual implemented system. (b) System operation diagram.*

rate. The experiments were carried out in the actual factory environment, with the participation of the workers operating the system. To evaluate the performance of each model, several commonly adopted metrics, which are highly relevant to the objectives and concerns of modern production strategies, were used, including processing time, sensitivity, specificity, accuracy, and defect rate.

The following sections will describe the development of an autonomous component testing system incorporating a vision system for improved reliability. Next, the utilization of the above-mentioned vision models on a specific dataset obtained from the autonomous component testing system will be demonstrated. The experimental results are analyzed and interpreted in detail with comparisons and discussions between the models based on several criteria.

2. AUTONOMOUS COMPONENT TESTING SYSTEM

We developed an autonomous camera module memory testing system, as depicted in Figure 1.

The system comprises a pick-and-place robot with a pneumatic nozzle to grip camera modules from a feeding tray and transfer them to a test socket (Figure 1a). The pick-and-place robot system can suffer from inaccuracies due to factors such as the tolerance of the feeding tray and the nonstandard behavior

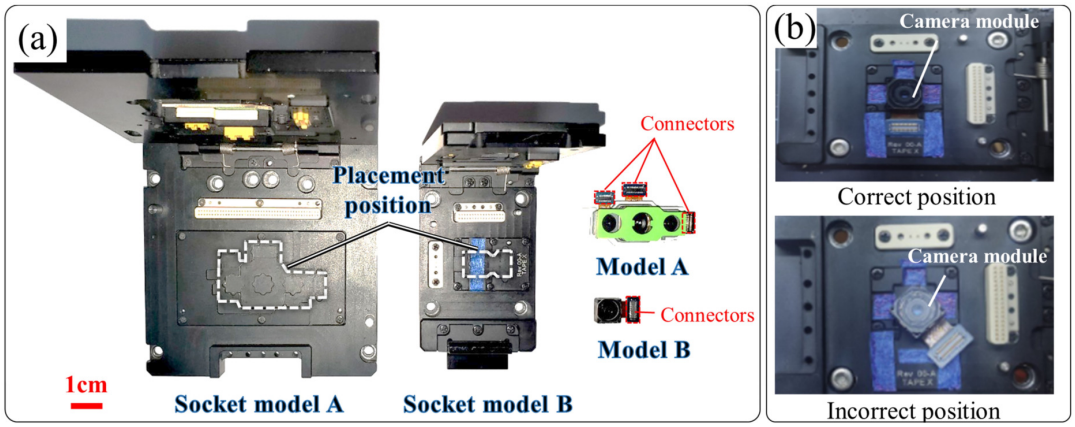


Fig.2: *Placement of camera modules on socket for electrical connecting and testing. (a) Socket and camera module of different versions. (b) Image of camera modules placed in the socket correctly and improperly.*

of factory workers, which can result in errors in the placement of modules. To enhance the system's reliability, we implemented a dual-camera setup, with one camera used to detect the presence of camera modules in the feeding tray and the other used to detect the position of the modules on the socket. The overall system can be divided into four sections, as depicted in Figure 1b. In Center Processing (Section A), a Programmable Logic Controller (PLC) is utilized to control the pneumatic valves, test socket, and pick-and-place robot in the operation area (Section C). An industrial computer (Section B) was used to acquire real-time images of the feeding tray and socket by connecting to the two cameras. The Control and Display Section (Section D) integrates HMI and LCD to enable efficient control and monitoring of the system.

In Section C, the test system includes a socket that makes good contact with the camera module and the testing circuit to perform the memory quality testing functions of the camera module. When the socket closes the lid, it connects to the camera module via connectors (Figure 2a). Since the connector's pins are tiny, even a slight misalignment between the camera module and the socket might cause severe damage. In this study, we focus on the Model A camera module, characterized by its small size and a single set of connectors. This model is lightweight, and its color is similar to the socket's, making it more difficult to recognize. The vision system equipped for the pick-and-place robot helps identify correctly and incorrectly positioned camera modules in the socket (Figure 2b), thus reducing the failure rate. A simple classification model was proposed based on computing the histogram of the image of the camera module on the socket, thereby detecting the incorrect position [34]. In this work, machine learning models are employed to improve the performance of the developed system.

3. DEPLOY MACHINE LEARNING AND COMPUTER VISION

3.1 Histogram analysis

Histogram Analysis presents an advantage as it offers a visual representation of data distribution, facilitating the identification of patterns and trends within the data set [37]. The procedure to detect incorrect placement of the camera module is depicted in the diagram in Figure 3 based on the analysis of the histogram of each image with the number of images of the input samples.

The average brightness value for each input image and the entire data set was computed to get the mean and variance. To determine the alignment of the camera module, the reference values are calculated using the following formulas:

$$m_i = \frac{sum(pixels\ value\ *\ number\ of\ pixel)}{number\ of\ pixel} \tag{1}$$

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} m_i\right) = \frac{m_1 + m_2 + m_3 + \ldots + m_n}{n} \tag{2}$$

$$s^2 = \frac{\sum_{i=1}^{n}(m_i - \bar{x})^2}{n-1} \tag{3}$$

$$CI = \bar{x} + z\frac{s}{\sqrt{n}} \tag{4}$$

Where m_i is the average value of the grayscale light intensity of the of the image, $\bar{x}$ is the average value over the entire dataset, n is the sample size, s^2 is the variance, the confidence interval (CI) and the confidence level z.

Three reference values $\bar{x}$, s^2, and CI will be used to determine the camera alignment based on a comparison algorithm shown in Figure 3. The algorithm aims to reduce processing time while maintaining efficiency through its simplicity.

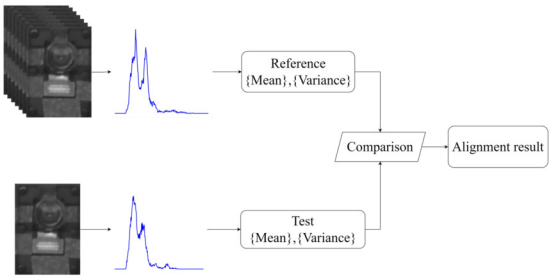


Fig.3: *Implementation diagram of the process of detecting camera misplacement based on histogram analysis.*

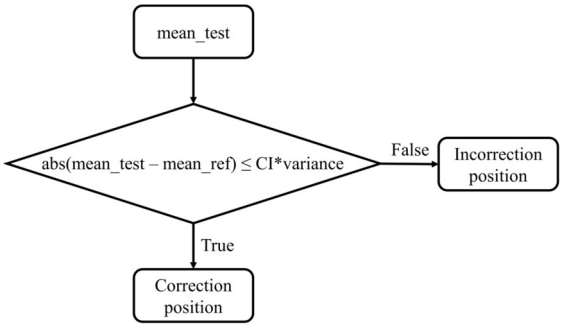


Fig.4: *Comparison conditions for alignment classification of camera module under test.*

3.2 Logistic Regression

Logistic Regression was found to be advantageous in our work due to its ability to accurately predict the class labels of the objects being analyzed based on a set of features and coefficients learned during the training phase [38]. The prediction model in this model has the basic form in which the continuous random variable will depend on the independent single or multiple variables. In our situation, the dependent variable is expressed by two statuses, which are the correct and incorrect position of the camera. At this point, an approximation using the binary logistic regression model was used to calculate the probability of the position of the camera module in the socket. This prediction process is based on the probability distribution $\pi(x)$, which has a value in the interval [0, 1] of the independent variables $x_1, x_2, \ldots, x_{n-1}, x_n$. These independent observations are the pixel values of the camera module image. The calibration function ensures that any value of x has a corresponding value of $\pi(x)$ in the range [0, 1]. The logistic transformation logit that performs the calibration is presented in Equation 5,

$$logit = \ln \frac{\pi(x)}{1 - \pi(x)} \tag{5}$$

Additionally, the logit function is also a representation of the independent variables such as:

$$logit = \omega_0 + \omega_0 x_1 + \cdots + \omega_n x_n \tag{6}$$

The weight vector is estimated using the maximum likelihood method, also called the training model process. From Equations 5 and 6, the probability distribution function $\pi(x)$ is shown as follows.

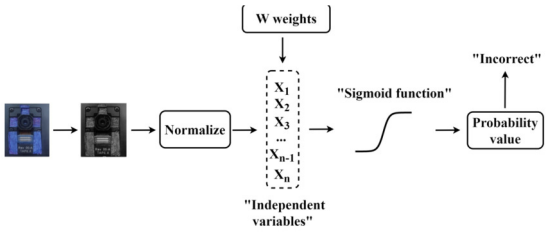
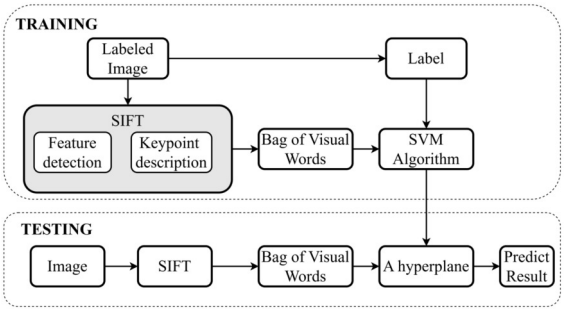


Fig.5: *The classification model using Logistic Regression.*

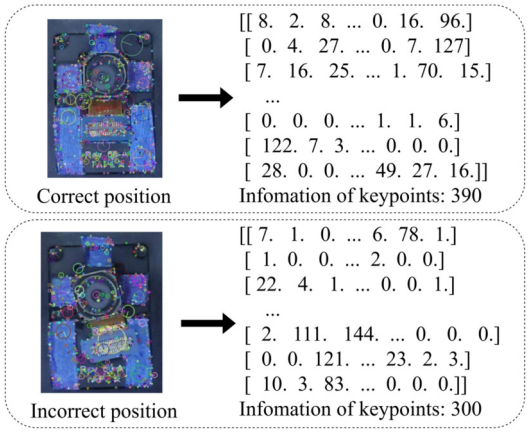
$$\pi(x) = \frac{\exp(\omega_0 + \omega_0 x_1 + \cdots + \omega_n x_n)}{1 + \exp(\omega_0 + \omega_0 x_1 + \cdots + \omega_n x_n)} \tag{7}$$

The form representing $\pi(x)$ in Equation 7 is also known as the Sigmoid function.

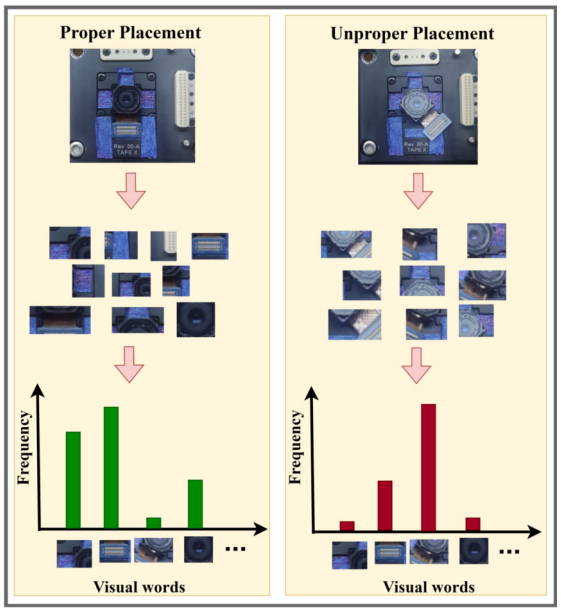
The process by which the model calculates and predicts the position of the camera module in the socket is shown in Figure 5. Each input data is transformed to grayscale and then normalized to obtain a vector of independent variables x. This value is multiplied by the weight vector according to Equation 6 and is evaluated by the sigmoid function in Equation 7. From the calculated probability, the state of the camera is determined.



(a)



(b)



(c)

Fig.6: *SVM classification process. (a) Implementation diagram. (b) Features extraction by SIFT on input data in case of proper placement and improper placement. (c) Vector quantization of two type of input data of proper and improper placement.*

3.3 SIFT-SVM

Support vector machine (SVM) is a supervised learning algorithm for classification, regression, and outlier detection problems. This was a popular algorithm until the dramatic development of neural network models. This algorithm works well for large data samples and often yields superior results compared to other algorithms in supervised learning [39], [40]. The idea of SVM is to find a hyperplane to separate the data layers. With datasets that do not have linear separation capabilities, SVM offers two solutions: Soft margin and kernel tricks [41]. In our problem, the RBF kernel was used to classify objects.

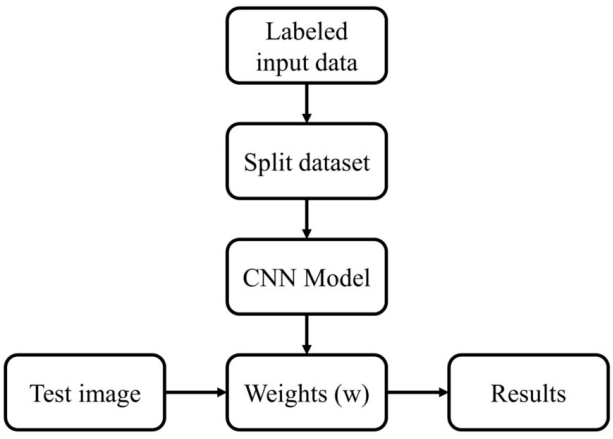


Fig.7: *Flowchart of the CNN model system.*

In our work, the input image must be preprocessed, extracted features, and then represented by a vector with the same dimensions. Here, the SIFT (scale-invariant feature transform) and BoVW (Bag of Visual Words) techniques were applied, as shown in Figure 6.

Features (key points) were extracted from camera module training images using the SIFT algorithm, and then each image can be represented by sets of key point descriptors (Figure 6b). The center of the circle is the location of the feature point. The line from the circle's center to the perimeter represents the gradient direction of the feature point. The length of the line represents the modulus of the gradient.

With the input data samples extracted, about 300 to 400 key points (this amount depends on the posture of the components on each data input). Each of these key points is characterized by a vector of 128. Consequently, using the input picture of 200 $\times$ 300 pixels and the Sift algorithm, the image maintains around 300 to 400 characteristics, each feature represented in 128-dimensional vectors.

Next, the BoVW - Bag of Visual Words was used to represent our data. The image is described by a histogram of quantized local features. More precisely, an unordered set of local patches is initially extracted and characterized by the SIFT technique in the preceding step. With many characteristics, the K-Mean Clustering algorithm was used to gather them into the most typical K clusters of the whole data set. In the problem of classifying this camera module, K = 200 was selected. Corresponding to the entire input data set, the center will select 200 cluster centers (keyword - 128-dimensional vector) to perform. After obtaining the dictionary (codebook), it is characterized by the entire input data set, and then each input image is compared to that dictionary.

Finally, after extraction characterized by SIFT, the input data is compared to the Codebook (in Figure 5c). From there, it will indicate the histogram chart frequency of visual keywords of each input data. This chart has a horizontal axis for the "visual words" and a vertical axis for the number of key points corresponding to each "visual word." So, corresponding to K = 200, each picture will be described by a vector with a dimension of 200. Each data now has an equal number of dimensions, although the characteristic number of each image is different from the other. This is useful for utilizing the Machine Learning algorithm in the following stage.

3.4 Deep learning models

Deep learning has gained widespread popularity and demonstrated remarkable performance in computer vision systems, especially concerning classification, object detection, and segmentation tasks. In this study, various types of deep learning models were utilized to assess classification performance, encom-

passing a simple CNN model as well as two YOLO classification variants: YOLOv8n and YOLOv8x.

Convolution Neural Network (CNN), also known as ConvNet, has a deep feedforward architecture and an astonishing capacity to generalize better than networks with fully connected layers. It can recognize objects effectively and learn highly abstract qualities [42], [43]. The convolutional neural network has been studied and implemented in the experimental task in order to evaluate and examine the results produced by this model due to its notable advantages (Figure 7). The input data were labeled according to the camera image in the first step. The data set is then split into three data sets for training, testing, and validation in the data set division procedure. The built-in CNN model was used to train these datasets.

Through the data exploration process, there was no environmental fluctuation because the data came from a fixed place. Furthermore, the more layers used, the longer the model takes to process. Therefore, in this paper, a common architecture was used, consisting of two layers of convolution, two layers of max–pooling, and a fully connected layer. Table 1 summarizes the settings we used for the convolution neural network.

Table 1: *Keras model summary.*

Layer (type)	Output Shape	Param♯
Conv2D	(None, 157, 122, 32)	896
MaxPooling2D	(None, 78, 61, 32)	0
Conv2D	(None, 78, 61, 32)	9248
MaxPooling2D	(None, 39, 30, 32)	0
Flatten	(None, 37440)	0
Dense	(None, 512)	19169792
Dense	(None, 1)	213

You Only Look Once (YOLO) introduces an end-to-end neural network that predicts both bounding boxes and class probabilities simultaneously. This stands in contrast to prior object detection algorithms, which repurposed classifiers to conduct detection as a separate step. YOLOv8 represents the most recent iteration featuring advanced object detection and image segmentation capabilities. Besides its proficiency in these domains, the model is equipped with inherent support for image classification [44]. Its noteworthy characteristics comprise a newly designed backbone network, facilitating straightforward comparisons with earlier YOLO models, the introduction of fresh loss functionalities, and incorporating an innovative anchorless detection head. We utilized two Classify models, namely YOLOv8n and YOLOv8x, both were pretrained on the COCO dataset [45], renowned for its extensive and diverse collection of images spanning various object classes.

4. EXPERIMENTAL RESULTS AND DISCUSSION

4.1 Experiments setup

The factory conducts a sampling process to ensure the real-life environment in which workers will operate the machine and the data collection process is automated (Figure 8). The sampling process for capturing images of camera modules in the autonomous system involves capturing images of the modules at various positions and angles. The input data sets are separated into two categories: the incorrect position image (IPI) data set and the correct position image (CPI) data set. To ensure that the obtained data is as close to reality as possible, workers naturally insert the feeding trays with camera modules into the system. In IPI cases, the transition error is within 0.5-2 mm from different directions, and the rotation error is within 0-12 degrees in both directions. In addition, cases with high error (2-10 mm and more than 12 degrees) are added to the data set to enrich the sample. In the CPI dataset, the camera modules are placed in the correct position, so the resulting samples are similar.

During the sampling process, we also vary the light levels on the subject with different intensities. In the system, we arrange LED strips for an average brightness of 100 lux, and the light levels are adjusted according to the bulb's power from 60-100% to collect samples. The data set is then manually labeled and

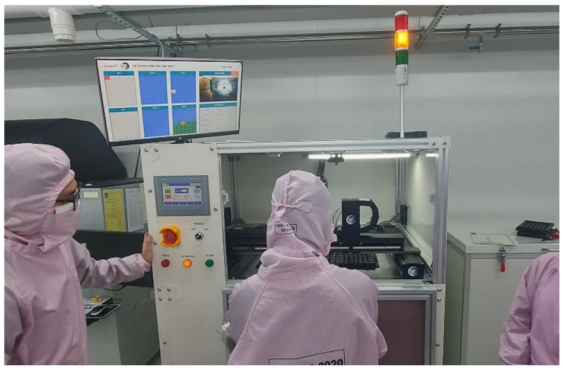


Fig.8: *Sampling process in the factory.*

divided into a training dataset and a testing data set. The training data set is used to train the model and contains 1250 CPI and 1114 IPI. The testing data set is used to evaluate the performance of the trained model and contains 1268 CPI and 1232 IPI. The main difference between the training dataset and the testing dataset is that the testing dataset is used to evaluate the model's performance on unseen data, whereas the training dataset is used to train the model.

The raw data captured consists of redundant regions, as in Figure 9. Leaving these image portions will decrease the accuracy of the models since they include a lot of extraneous information, and the camera module is too small to discriminate between Correct

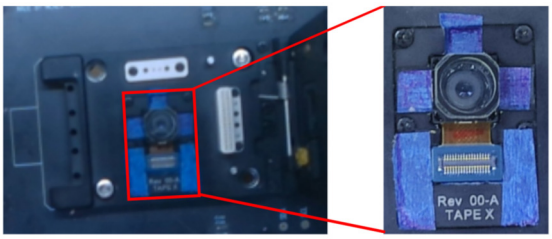


Fig.9: *Sample image after processing.*

and Incorrect positions. The initially captured image is cropped to retain the area of the camera module, and other socket areas are removed (in Figure 9). The input data for the machine learning models under consideration is an image with a consistent size across all models. However, a notable differentiation exists in the color representation of images used by different models. Specifically, the CNN, YOLOv8, and SVM employ RGB images, while the Logistic Regression and Histogram models rely on grayscale images for their computations and analysis. This choice in input data representation is motivated by the specific feature extraction techniques employed by each model.

In the case of a CNN or an SVM model, the input image is typically a color image represented in the RGB (red, green, blue) format. This format allows the model to capture the full range of colors present in the image, which can be important for image classification and recognition tasks. When it comes to detecting incorrect positions of camera modules in a pick-and-place system, color information can provide valuable features for the model to identify subtle differences in the position and orientation of the modules. CNNs are designed to automatically extract features from images using convolutional layers, which apply filters to the input image to identify and extract features such as edges, shapes, and textures. These features can then be passed to the fully connected layers in the model for classification. SVM models can also be used for image classification, but SIFT key points are typically used to extract features from the image and build a histogram of these features, which is then used as input to the SVM.

For Histogram and Logistic Regression models, the input image is typically a grayscale image. A grayscale image is an image representation in which each pixel is represented by a single value, representing its intensity or brightness rather than its color. In this representation, the color information is discarded, and the image is transformed into a single-channel image, where the intensity of each pixel ranges from 0 to 255. In the case of histogram models, the input image is transformed into a histogram representation, where the image is divided into several bins, and the number of pixels in each bin is counted. The resulting histogram represents the distribution of intensity values in the image, which can be used as features for classification. In terms of Logistic Regression models, the grayscale image is directly used as input features for classification. Logistic regression models are linear models that are widely used for binary classification tasks.

4.2 Results and Discussion

The models are evaluated according to the following criteria: processing time, accuracy, sensitivity, specificity, and defect rate. It is imperative to emphasize that the industrial computers utilized in this study do not incorporate a Graphics Processing Unit (GPU) to enhance the computational prowess of machine learning models. As a consequence, all tasks are executed solely on the available Central Processing Unit (CPU). Therefore, the execution time of the models will inevitably differ when employing systems equipped with GPU computing capabilities. The processing time for each model is calculated from the instance in which the cropped image is introduced to the model until the resulting output is obtained. This computation comprises the duration necessary for preprocessing the cropped image to conform to the model input (if applicable) and the inference time required to produce prediction outcomes. First, we define some concepts as follows:

About True Label:

- Correct placement of camera – 0
- Incorrect placement of camera – 1

About Predicted Label:

- Predict that the camera is misaligned, equivalent to 1
- Predict that the camera is aligned, equivalent to 0

At that time, the precise level measurement parameters, including True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), are defined as follows in Table 2. The results of measuring parameters according to the data set are presented in Figure 10.

Table 2: *The precise level measurement parameters.*

	True label	**Predicted label**
TP	1	1
TN	0	0
FP	0	1
FN	1	0

The accuracy of the model was calculated by:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{8}$$

Sensitivity indicates the probability that the model correctly predicted the true positive prediction. The higher the percentage, the smaller the FN, and the lower the chance of misjudging.

$$Sensitivity = \frac{(TP)}{(TP + FN)} \tag{9}$$

The specificity represents the probability that the model correctly predicts a true negative prediction. The higher the percentage, the smaller the FN, and the smaller the chance of missing the right camera.

$$Specificity = \frac{(TN)}{(TN + FP)} \tag{10}$$

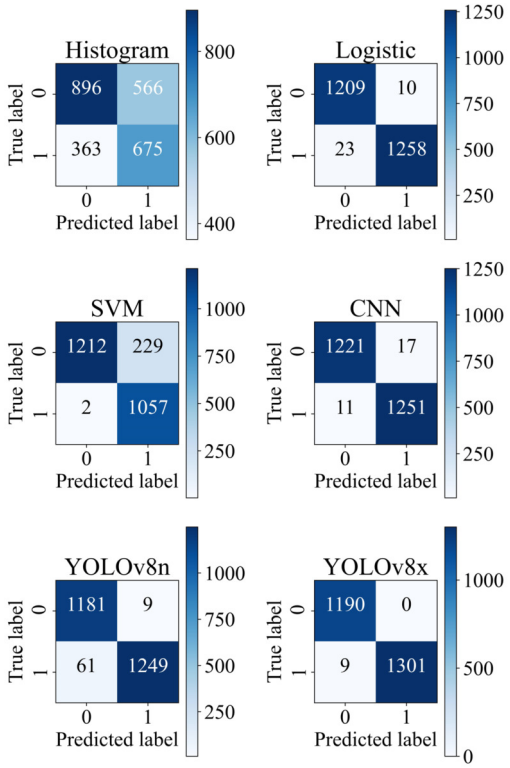
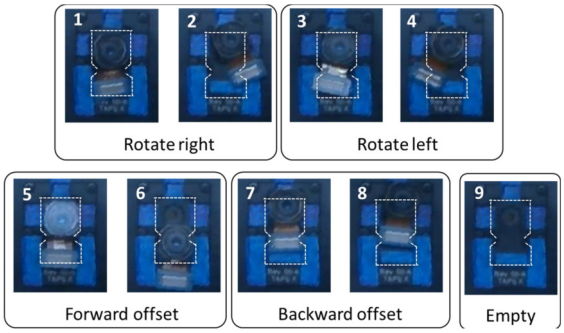


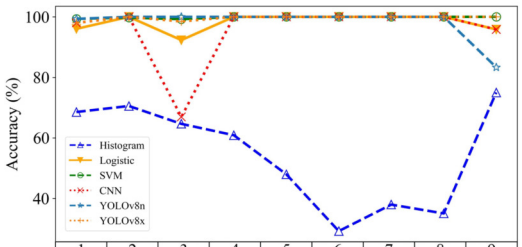
Fig.10: *Confusion matrix of 6 models.*

Additionally, we evaluated these models based on the offset of various modules, considering both the translational and rotational deviations. For our evaluation, we selected images from the test data set and divided them into nine deviation categories, as shown in Figure 11a. The categories included four levels of rotational deviation, four levels of translational deviation, and an empty case. The closer the camera module is to its correct placement in the socket, the harder it is to detect. The results from the logistic, SVM, and Deep learning models indicate relatively high accuracy in most cases (Figure 11b); however, in Case 3, the CNN model performed poorly with an accuracy rate of only 66.92%. The SVM model exhibited exceptional results with an accuracy greater than 99% in all cases, while the histogram analysis model performed the worst with accuracy rates below 75% and reaching as low as 29%.

Finally, we evaluated models using the complete test data set and presented the results using the cri-



(a)



	1	2	3	4	5	6	7	8	9
Histogram	68.54	70.53	64.62	60.87	47.95	29.17	37.93	35.0	75.0
Logistic	96.03	100.0	92.31	100.0	100.0	100.0	100.0	100.0	95.83
SVM	99.34	99.74	99.23	100.0	100.0	100.0	100.0	100.0	100.0
CNN	98.01	100.0	66.92	100.0	100.0	100.0	100.0	100.0	95.83
YOLOv8n	99.34	100.0	100.0	100.0	100.0	100.0	100.0	100.0	83.33
YOLOv8x	98.01	100.0	98.46	100.0	100.0	100.0	100.0	100.0	100.0

(b)

Fig.11: *The Accuracy of Models for Specific Test Cases.*

teria outlined in Table 3. The histogram model, when applying smaller camera data, lost its accuracy rate to 99.92%, which dropped to just 62.84%. Therefore, the histogram model is no longer viable for the present problem and future versions with complex components. The YOLOv8x model gives outstanding results when the accuracy is up to 99.64%. The three models, Logistic Regression, CNN, and YOLOv8x, also bring high accuracy of 98.68%, 98.88%, and 99.64%, respectively. However, the CNN model has a recognition time performance of up to 0.34959 seconds, which is a huge processing time result for an automatic system that operates in a factory (Figure 12). In contrast, the rest of the models have very small processing times of 0.00049 seconds – Logistic Regression model, 0.00091 seconds – histogram model, 0.01280 seconds - YOLOv8n, 0.01578 seconds – SVM model and 0.04660 seconds - YOLOv8x.

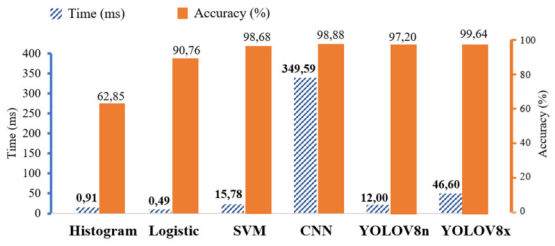


Fig.12: *The accuracy and processing time of models.*

Table 3: *Comparative indicators in the models.*

Model	Time(s)	Sensitivity	Specificity	Accuracy
Histogram	0.00091	0.65029	0.61286	0.62840
Logistic	**0.00049**	0.98205	0.99719	0.98680
SVM	0.01578	**0.99811**	0.84108	0.90760
CNN	0.34959	0.99107	0.98627	0.9880
YOLOv8n	0.01280	0.95344	0.99244	0.97200
YOLOv8x	0.04660	0.99313	**1.00000**	**0.99640**

In terms of sensitivity and specificity, the results obtained using histogram analysis are always significantly lower than those of the other machine learning models. Regarding the specificity of the system, the YOLOv8x model was found to show better results (1.0000) since the lowest camera omission rate was set correctly when using the model. The Sensitivity characterizes the proportion of the visual system that mistakenly predicts that the camera module is set from incorrect to correct. When such a phenomenon occurs, the socket is still closed to check the components, leading to the camera being stamped and causing destruction. Therefore, the sensitivity of the system is also the result of the evaluation of the defect rate according to the Sigma method (Figure 13). As a result, the application of the SVM model gives the best results (99.811%). The SVM provides a camera failure rate of only about 1888 PPM, equivalent to the number of misplaced cameras damaged by the socket.

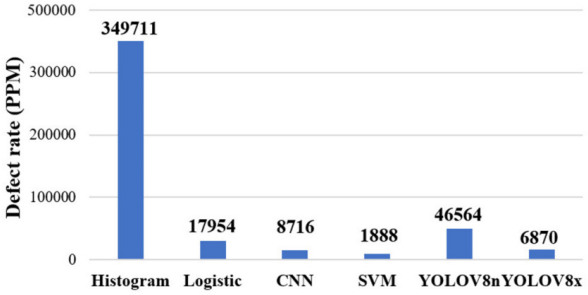


Fig.13: *The defect rate of all models.*

It is clear that Sift-SVM is the most appropriate machine learning model to improve the reliability of the automatic defect detection system. The results of the experiments, which were conducted in a practical factory environment with the participation of workers operating the system, showed that Sift-SVM had a processing time of 0.01578 seconds and the highest sensitivity of 99.811%. Furthermore, the model effectively reduced the failure rate to 1888 PPM, highlighting its advantages in terms of speed and accuracy. The results of this study have significant implications for the use of computer vision-based models in modern manufacturing. Demonstrates that automatic defect detection systems can significantly improve traditional manual detection methods, reducing the defect rate and increasing efficiency.

These findings support the use of Sift-SVM in practical industrial applications and suggest that further research should be conducted to explore its potential for widespread adoption in the industry. The potential for future improvement in the reliability of machine vision systems is immense as advances in technology and increasing computing power continue to drive the field forward. The versatility of the system is demonstrated by its compatibility with various camera module models and its ability to integrate with different types of robotic systems, highlighting its potential for widespread application in diverse industrial settings. The results obtained also highlight the importance of continuous research and innovation in automatic inspection detection and its role in improving the competitiveness of modern manufacturing.

5. CONCLUSION

This paper evaluated computer vision models, including Histogram analysis, Logistic Regression, SVM, and Deep learning, to detect camera module placement deviations. The effectiveness of the system is verified by several evaluation criteria, that is, time processing, accuracy, sensitivity, specificity, and defect rate. The results demonstrate that the Sift-SVM model proved to be the most effective in detecting improper placement of camera modules in the camera socket of the testing system. The SVM model outperformed the other models with accuracy scores above 99% for all test cases. The results of this study provide valuable insight into the optimization of automatic defect detection technology, which has emerged as a crucial area of research in modern manufacturing. The experiments carried out in an actual factory environment with the participation of the workers operating the system demonstrated the relevance of the selected metrics, including processing time, sensitivity, specificity, accuracy, and defect rate, to the objectives and concerns of contemporary production strategies. Implementing the Sift-SVM model can improve the reliability of the pick-and-place robot and decrease the system's defect rate, thereby achieving a higher Sigma standard and better overall performance.

References

[1] Y. Chen, X. Peng, L. Kong, G. Dong, A. Remani, and R. Leach, "Defect inspection technologies for additive manufacturing," *International Journal of Extreme Manufacturing*, vol. 3, no. 2, p. 22002, 2021.

[2] Z. Ren, F. Fang, N. Yan, and Y. Wu, "State of the art in defect detection based on machine vision," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 9, no. 2, pp. 661–691, 2022.

[3] N. N. S. A. Rahman, N. M. Saad, A. R. Abdullah, and N. Ahmat, "A review of vision based defect detection using image processing techniques

for beverage manufacturing industry," *Jurnal Teknologi*, vol. 81, no. 3, pp. 33–47, 2019.

[4] C. Vedang and S. Brian, "A Comparative Study of Machine Vision Based Methods for Fault Detection in an Automated Assembly Machine," *Procedia Manufacturing*, vol. 1, pp. 416–428, 2015.

[5] Z. Chen, J. Deng, Q. Zhu, H. Wang, and Y. Chen, "A Systematic Review of Machine-Vision-Based Leather Surface Defect Inspection," *Electronics (Basel)*, vol. 11, no. 15, 2022.

[6] S. Qi, J. Yang, and Z. Zhong, "A Review on Industrial Surface Defect Detection Based on Deep Learning Technology," in *Proceedings of the 2020 3rd International Conference on Machine Learning and Machine Intelligence, in MLMI '20. Association for Computing Machinery*, pp. 24–30, 2020.

[7] K. Christian, G. Kristina, K. Varun, A. Burcu, and F. Paul, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Advanced Engineering Informatics*, vol. 29, no. 2, pp. 196–210, 2015.

[8] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[9] P. V. Arun, "A CNN based Hybrid approach towards automatic image registration," *Geodesy and Cartography*, vol. 39, no. 3, pp. 121–128, 2013.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] P. Jiang, D. Ergu, F. Liu, Y. Cai, and B. Ma, "A Review of Yolo algorithm developments," *Procedia Computer Science*, vol. 199, pp. 1066–1073, 2022.

[13] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," *Intelligent computing: image processing based applications*, pp. 1–16, 2020.

[14] W. Zhiqiang and L. Jun, "A review of object detection based on convolutional neural network," in *2017 36th Chinese control conference (CCC)*, pp. 11104–11109, 2017.

[15] H. Wang and D. Hu, "Comparison of SVM and LS-SVM for regression," in *2005 International conference on neural networks and brain*, pp. 279–283, 2005.

[16] S. V. M. Vishwanathan and M. N. Murty, "SSVM: a simple SVM algorithm," in *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No. 02CH37290)*, pp. 2393–2398, 2002.

[17] Q. Li and X. Wang, "Image classification based on SIFT and SVM," in *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*, pp. 762–765, 2018.

[18] Y. Liu, S. Zhou, and Q. Chen, "Discriminative deep belief networks for visual data classification," *Pattern Recognit*, vol. 44, no. 10–11, pp. 2287–2296, 2011.

[19] Y. Hua, J. Guo, and H. Zhao, "Deep belief networks and deep learning," in *Proceedings of 2015 International Conference on Intelligent Computing and Internet of Things*, pp. 1–4, 2015.

[20] M. Robotyshyn, M. Sharkadi, and M. Malyar, "Surface defect detection based on deep learning approach," in *IntSol Workshops*, pp. 32–44, 2021.

[21] M. Baygin, M. Karakose, A. Sarimaden, and E. Akin, "Machine vision based defect detection approach using image processing," in *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pp. 1–5, 2017.

[22] S. Panagiotis, P. Alexios, and P. Dimitris, "A vision-based system for real-time defect detection: a rubber compound part case study," *Procedia CIRP*, vol. 93, pp. 1230–1235, 2020.

[23] S. Barua, F. Liou, J. Newkirk, and T. Sparks, "Vision-based defect detection in laser metal deposition process," *Rapid Prototyping Journal*, vol. 20, no. 1, pp. 77–85, 2014.

[24] J. P. Yun, S. Choi, and S. W. Kim, "Vision-based defect detection of scale-covered steel billet surfaces," *Optical Engineering*, vol. 48, no. 3, p. 37205, 2009.

[25] Y. Li, H. Wang, L. M. Dang, H.-K. Song, and H. Moon, "Vision-Based Defect Inspection and Condition Assessment for Sewer Pipes: A Comprehensive Survey," *Sensors*, vol. 22, no. 7, p. 2722, 2022.

[26] Y. He, K. Song, Q. Meng, and Y. Yan, "An End-to-End Steel Surface Defect Detection Approach via Fusing Multiple Hierarchical Features," in *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 4, pp. 1493–1504, 2020.

[27] J. Zhong, Z. Liu, Z. Han, Y. Han, and W. Zhang, "A CNN-Based Defect Inspection Method for Catenary Split Pins in High-Speed Railway," in *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 8, pp. 2849–2860, 2019.

[28] R. Shanmugamani, M. Sadique, and B. Ramamoorthy, "Detection and classification of surface defects of gun barrels using computer vision and machine learning," *Measurement*, vol. 60, pp. 222–230, 2015.

[29] H.-D. Lin and H.-L. Chen, "Automated visual

fault inspection of optical elements using machine vision technologies," *Journal of Applied Engineering Science*, vol. 16, no. 4, pp. 447–453, 2018.

[30] Ç. Aytekin, Y. Rezaeitabar, S. Dogru, and I. Ulusoy, "Railway Fastener Inspection by Real-Time Machine Vision," in*IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1101–1107, 2015.

[31] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using deep learning to detect defects in manufacturing: a comprehensive survey and current challenges," *Materials*, vol. 13, no. 24, p. 5755, 2020.

[32] T. Wang, Y. Chen, M. Qiao, and H. Snoussi, "A fast and robust convolutional neural network-based defect detection model in product quality control," *The International Journal of Advanced Manufacturing Technology*, vol. 94, pp. 3465–3471, 2018.

[33] Y. Yang *et al.*, "A High-Performance Deep Learning Algorithm for the Automated Optical Inspection of Laser Welding," *Applied Sciences*, vol. 10, no. 3, p. 933, 2020.

[34] H.-A. Phan *et al.*, "Development of a Vision System to Enhance the Reliability of the Pick-and-Place Robot for Autonomous Testing of Camera Module used in Smartphones," in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1–6, 2021.

[35] H. Wang, "A Review of Six Sigma Approach: Methodology, Implementation and Future Research," in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, 2008.

[36] E. Atmaca and S. S. Girenes, "Lean Six Sigma methodology and application," *Qual Quant*, vol. 47, pp. 2107–2127, 2013.

[37] C. Bradley and Y. S. Wong, "Surface texture indicators of tool wear-a machine vision approach," *The International Journal of Advanced Manufacturing Technology*, vol. 17, pp. 435–443, 2001.

[38] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*, vol. 398. John Wiley & Sons, 2013.

[39] B. C. Ko, K.-H. Cheong, and J.-Y. Nam, "Fire detection based on vision sensor and support vector machines," *Fire Safety Journal*, vol. 44, no. 3, pp. 322–329, 2009.

[40] E. Hortal *et al.*, "SVM-based brain–machine interface for controlling a robot arm through four mental tasks," *Neurocomputing*, vol. 151, pp. 116–121, 2015.

[41] M. N. Murty, R. Raghava, M. N. Murty, and R. Raghava, "Kernel-based SVM," *Support vector machines and perceptrons: learning, optimization, classification, and application to social networks*, pp. 57–67, 2016.

[42] Z. Zhang, "Derivation of backpropagation in convolutional neural network (cnn)," *University of Tennessee, Knoxville, TN*, vol. 22, p. 23, 2016.

[43] A. Mikołajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *2018 international interdisciplinary PhD workshop (IIPhDW)*, pp. 117–122, 2018.

[44] S. Akhtar, M. Hanif, and H. Malih, "Automatic Urine Sediment Detection and Classification Based on YoloV8," in *International Conference on Computational Science and Its Applications*, pp. 269–279, 2023.

[45] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, 2014.

Hoang Anh Phan received his BS degree in electronic engineering from Post and Telecommunication Institute of Technology, Hanoi, Vietnam in 2019, and master degree in electronic engineering from VNU University of Technology, Hanoi, Vietnam, in 2021. He is currently a lecturer at Faculty of Electronic Technology, VNU, University of Technology, Hanoi, Vietnam. His main research interests include robotics, automation and BioMems.

Van Tan Duong received the BS degree in Robotics Engineering from the University of Engineering and Technology - Vietnam National University Hanoi (VNUH) in 2023. He is currently working toward the MSc degree in Electronic Engineering at the University of Engineering and Technology - VNUH. His main research focuses on robotics, computer vision, and machine learning.

Mai Nguyen Thi received a B.E. degree in Robotics Engineering at University of Engineering and Technology, VNU, Hanoi, in 2023. Currently, she is working as an AI Engineer at GPS Vietnam Trading Limited Company. Her research interests include deep learning, computer vision, image processing, and face solutions.

Anh Nguyen Thi received a B.E. degree in Robotics Engineering at University of Engineering and Technology, VNU, Hanoi, in 2023. Now, she is an Artificial Intelligence Researcher at Viettel Artificial Intelligence & Data Services Center. Her research interests include machine learning, deep learning and biomedical.

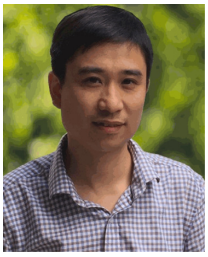
Hang Khuat Thi Thu received a B.E. degree in Robotics Engineering at University of Engineering and Technology, VNU, Hanoi, in 2023. Now, she is a Master's student in Electronic Engineering at the University of Engineering and Technology, VNU, Vietnam. Her research interests include robotics UAVs and swarm optimal.

Thang Luu Duc received his B.E. degree in Robotics Engineering at the University of Engineering and Technology, VNU, Hanoi, in 2023. He is currently a Computer Vision Engineer, conducting research and integrating AI technology into the industrial field at CAE Technology company, Vietnam. His research interests include deep learning, image processing, and Optical Character Recognition (OCR).

Van Hieu Dang received his BS degree in robotics engineering from University of Engineering and Technology, Hanoi, Vietnam in 2023. Now, he is a Master's student in electronic engineering, VNU, University of Technology, Hanoi, Vietnam. His main research interests include robotics and automation.

Huu Quoc Dong Tran received a B.E. degree in Robotics Engineering at University of Engineering and Technology, Vietnam National University, Hanoi, in 2023. Now, he is a Master's student in Systems and Control Engineering at the Tokyo Institute of Technology, Japan. His research interests include control theories, motion planning, localization, and robotics vision.

Thi Thanh Van Nguyen gained B.Sc in Electronics and Telecommunication from Vietnam National University, Hanoi in 2001 and M.S in Mechatronics from Asia Institute of Technology, Thailand in 2006. She now is teaching in Electronics Engineering field in University of Engineering and Technology Vietnam National University, Hanoi. Her research interests are in the area of stable control, localization and navigation for autonomous mobile robot system.



Thanh Tung Bui received the BS degree in electrical engineering from Vietnam National University, Hanoi (VNUH) in 2004, and the ME and DEng degrees in science and engineering from Ritsumeikan University, Shiga, Japan, in 2008 and 2011, respectively. From 2011 to 2015 he was a postdoctoral researcher with the 3D Integration System Group, Nanoelectronics Research Institute (NeRI), National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. Currently, he is an associate professor in the Faculty of Electronics and Telecommunication (FET), VNU University of Engineering and Technology (UET), Vietnam National University, Hanoi (VNUH). His current interests include microfluidics, MEMS-based sensors, actuators and applications. He is the author and co-author of more than 100 scientific articles and 7 inventions.