

Support vector machine-based object classification for robot arm system

Vo Duy Cong¹, Thai Thanh Hiep²

¹Industrial Maintenance Training Center, Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam

²Automotive Engineering, Faculty of Transportation Engineering, Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam

Article Info

Article history:

Received Sep 7, 2022

Revised Mar 10, 2023

Accepted Mar 28, 2023

Keywords:

Classification

Raspberry Pi

Robot arm

Sorting system

Support vector machine

ABSTRACT

In this paper, a support vector machine (SVM) model is trained to classify objects in the automatic sorting system using a robot arm. The robot arm is used to grab objects and move them to the right position according to their shape predicted by the SVM model. The position of objects in the image is identified by using the contouring technique. The centroid of objects is calculated from the image moment of the object's contour. The calibration is conducted to get the parameters of the camera and combine with the pinhole camera model to compute the 3D position of the objects. The feature vector for SVM training is the zone feature and the SVM kernel is the Gaussian kernel. In the experiment, the SVM model is used to classify four objects with different shapes. The results show that the accuracy of the SVM classifier is 99.72%, 99.4%, 99.4% and 99.88% for four objects, respectively.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Vo Duy Cong

Industrial Maintenance Training Center, Ho Chi Minh City University of Technology

268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

Email: congvd@hcmut.edu.vn

1. INTRODUCTION

The continuous development of science and technology has had a great impact on people's lives and working habits. People are increasingly demanding more about the working environment, income and other social benefits. Automated machines have gradually replaced humans to perform dangerous, toxic or repetitive boring jobs. In particular, automation systems using robotic arms have been strongly developed in recent years. Robotic arms with flexible movement, tireless continuous operation have been applied in many different industries. Robotic arms can perform both tasks that humans will not undertake and tasks that humans cannot [1]–[3]. The use of robotic arms to replace humans has brought many benefits such as increasing product quality, productivity, reducing material waste, and costs [4]–[8].

Nowadays, with the support of sensor technology, especially vision sensors, the robot's functionality has been expanded [9], [10]. Robots not only execute pre-programmed tasks in a stationary environment, but they can also execute different tasks in unstructured environments by using sensors to sense their surroundings. Visual sensors with the ability to extract complex information have been widely in robot systems and machine vision-based robotic manipulators have been employed recently in automatic systems [11]–[14]. The computer vision system provides the necessary information for robots to respond to changes in the environment [15], [16]. It captures images of the target objects and processes images by using algorithms to extract data from images. These data are transmitted to the robot system. Then, the robot system uses this data to control the motion of robot manipulators and other devices [17]–[20]. For example, in an automatic pick and place system, the camera will capture images of some objects in the workspace. The images are processed to determine the

coordinates of objects. These coordinates are transmitted to the robot system. The robot arm is controlled to the objects' position and grasps objects to another position. There have been many other studies that have used vision systems in robotic systems for many industrial automation applications.

The problems of detection, identification and classification are very common in automation production [21], [22]. Using cameras for classification has provided benefits such as high repeatability, consistent accuracy, and high speed of classification. Many image processing algorithms have been proposed for classifying different objects. Support vector machine (SVM) is a machine learning algorithm that is widely used for classification applications [23]–[26]. Many studies have revealed that the SVM algorithm can perform better than an artificial neural network (ANN) [27]. In this paper, we train an SVM model to classify objects in an application using a robot arm to sort objects. The system consists of a 3-DOF robot arm to grab objects, and a camera to capture images of objects. The SVM is trained on a personal computer and the trained model is run on a Raspberry Pi computer. The SVM will classify objects with different shapes. To train an SVM model for classification, we took images of the objects and processed them to obtain a dataset. The dataset is expanded by using a rotation augmentation to improve the accuracy and generalization of the SVM model. The rest of the paper will describe in detail the robot system and the SVM algorithm for object classification.

2. RESEARCH METHOD

2.1. Proposed system

This paper uses a developed robot arm as shown in Figure 1. The hardware and software to build the robot arm are described as follows: The hardware system composes a 3-DOF robot arm for grabbing objects, a camera fixed to take pictures of objects, and a Raspberry Pi computer to process images and classify using SVM model. Three stepper motors mount at each joint of the robot arm to provide the rotation movement for links. The motors are controlled by an Arduino Uno board. The Arduino board sends the pulses and direction signals to stepper motor drivers. These drivers convert the signals from Arduino to the current signals to rotate the motors.

- Stepper motors: the stepper motors are bipolar stepper types equipped with a high-precision planetary gearbox with a gear ratio of 5:1. It can provide a maximum holding torque of 30 kgf.cm. The resolution of the motor is 1.8 degrees. The dimension of the motors is 42×42×97 mm.
- The A4998 drivers: The A4998 is used to control the small-size stepper motors. The supply voltage for the driver is from 8 to 35 V, the maximum output current to each phase of motor is 2 A. It supports multiple operating modes of bipolar stepper motor with five different step resolutions: full-step, 1/2, 1/4, 1/16, and 1/32. Figure 2 shows the motor and driver used to build the robot arm, Figure 2(a) is the Nema 17 motor with a gearbox and Figure 2(b) is the A4998 driver.



Figure 1. Robot arm

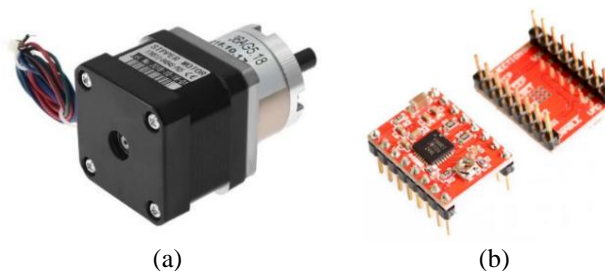


Figure 2. Stepper motor and A4998 driver: (a) stepper motor and (b) A4998 driver

- Arduino Uno R3: it is based on the ATmega328P which is a mid-range microcontroller with 8-bit registers. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator. Arduino Uno is suitable for many embedded applications due to the variety of library support for devices such as sensors, motors, and communication.
- Camera: The camera is a USB camera with a 30 W pixel sensor for high-quality images. It has the function of backlight compensation and strong light compensation. The frame frequency is 30 FPS at a resolution of 640×480 pixels. It supports a USB 2.0 output to connect to a computer.
- Raspberry Pi 4 computer: The Raspberry Pi 4 is the latest version of the Raspberry Pi family and brings with it a 1.5GHZ quad-core ARM Cortex-A72 for better performance. There are also more RAM options for this Raspberry Pi. It has 40 pins extended GPIO to connect with other devices. We can use it as a small computer to do most things a desktop computer can do. The main operating system for the Raspberry Pi is Raspbian which is based on Debian.

For writing programs to control the operation of the system, we utilized some software tools as follows:

- Arduino IDE: is a software to write firmware and send it to Arduino boards. The language for writing is C/C++. We can install many libraries that have been developed by other programmers to the Arduino IDE. In our program, we only include a header file to use these libraries. In this work, we use the latest version of Arduino IDE 1.8.19 to build the code for controlling the robot arm.
- Python 3.9: Python is a popular general-purpose programming language. It is utilized for many applications such as desktop applications, web development, machine learning, and many other fields. Many libraries for image processing and machine learning can be installed in the Python environment. We use this programming for processing images to extract features and train the SVM model.
- OpenCV library: OpenCV is an open-source library for Computer Vision and Machine Learning. OpenCV now supports a multitude of algorithms that is easy to implement. OpenCV-Python is a library of Python bindings designed to solve computer vision problems.

2.2. Support vector machine

Support vector machines (SVMs) are a well-known supervised machine learning algorithm for classification and regression. It has been shown to be highly accurate and generalizable compared to other data classification algorithms. SVMs were first presented by Vapnik and have since been applied in many applications thank to many attractive features and high empirical performance.

SVMs are a binary classification that separates data into two classes by a hyperplane plane. Two parallel margin hyperplanes lying on each side of the hyperplane that separates the data are constructed. The data belongs to only one side of these two planes. The optimal position of the hyperplane for separating data is determined where the distance between two margin hyperplanes is maximum. For complex data sets, the data is mapped to a higher dimensional space using kernel functions and constructs a hyperplane in this space to separate data.

Given a set of m data points $(x_i, y_i), i = 1 \div m, x_i \in R^n$ is the input vector, $y_i \in \{-1,1\}$ is the output classification. The SVMs are required to solve the optimization problem defined as (1) [28]:

$$\begin{aligned} \min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^m \xi_i \\ \text{Subject to } y_i(w \cdot x_i + b) + \xi_i \geq 1 \end{aligned} \quad (1)$$

where w is the weight vector, b is the bias, ξ_i is the error for a given training point x_i , C is a constant to adjust the error. The optimal problem of SVMs is reformulated into the dual formulation by using the Lagrange method [29].

$$\begin{aligned} \max_{\lambda_i} \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \lambda_i \lambda_j y_i y_j (x_i^T x_j) \\ \text{Subject to } \sum_{i=1}^m \lambda_i y_i = 0, 0 \leq \lambda_i \leq C \end{aligned} \quad (2)$$

If the data x_i is mapped into a higher dimensional space by the function ϕ and $K(x_j, x_i) = \phi^T(x_j)\phi(x_i)$ is the Kernel function, we obtain the nonlinear support vector machine [29].

$$\begin{aligned} \max_{\lambda_i} \sum_{i=1}^m \lambda_i - \frac{1}{2} \sum_{i=1}^m \sum_{j=1}^m \lambda_i \lambda_j y_i y_j K(x_j, x_i) \\ \text{Subject to } \sum_{i=1}^m \lambda_i y_i = 0, 0 \leq \lambda_i \leq C \end{aligned} \quad (3)$$

There are many kernel functions in SVMs, and the selection of kernel function have a significant effect on the performance of SVMs. There are some popular kernel functions [22].

- Linear Kernel

$$K(x_j, x_i) = x_j^T x_i \quad (4)$$

- Polynomial kernel

$$K(x_j, x_i) = (\gamma x_j^T x_i + \tau)^p \quad (5)$$

- Gaussian RBF kernel

$$K(x_j, x_i) = \exp\left(-\frac{1}{2\sigma^2} \|x_j - x_i\|^2\right) \quad (6)$$

- Sigmoid kernel

$$K(x_j, x_i) = \tanh(\beta_0 x_j^T x_i + \beta_1) \quad (7)$$

To apply SVM for multi-classification, we can use either a multi-class version of the SVM algorithm or combine binary classifiers using a decision function. The common methods for multi-class SVM are one vs one (OvO), one vs all (OvA), direct acrylic graph (DAG), error-correcting output codes (ECOC), and Binary tree architecture (BTA) [29]. In this paper, we will use SVM to classify multiple objects and the method One vs one (OvO) is chosen.

2.3. Implement support vector machine for objects classification

Figure 3 is the steps to implement the SVM algorithm for classifying objects. First, the image of objects is taken by the camera in the RGB color space. It is transformed to a grayscale image. The image is then filtered noise before converting to a binary image. Gaussian filter is used to remove high-frequency content (e.g., noise and edges) from the grayscale image. A fixed threshold is used to convert the grayscale image to a binary image. If the gray value of a pixel exceeds a threshold, then the new pixel value is set to 1. Otherwise, the pixels are set to 0. By choosing an appropriate threshold, the objects in the binary image are white blobs. We identify these blobs by using contouring technology. Contouring is a method that simply joins all continuous points of the same color and intensity [20]. The image moment of the contour can be calculated to determine its centroid:

$$\begin{aligned} c_x &= \frac{m_{10}}{m_{00}} \\ c_y &= \frac{m_{01}}{m_{00}} \end{aligned} \quad (8)$$

where (c_x, c_y) is the centroid of the object, m_{ij} is the image moment. OpenCV library has defined functions to find the contour of an object and calculate its image moment.

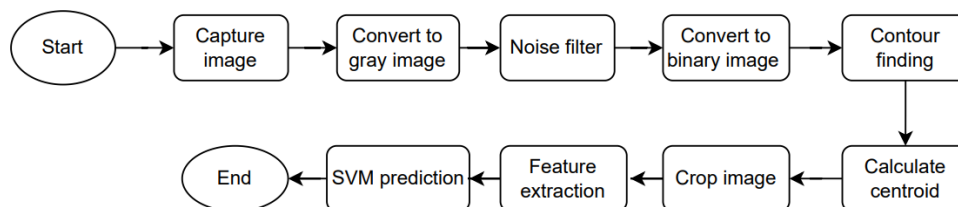


Figure 3. Flow chart of classification algorithm

After determining the centroid of the object, image cropping is performed to extract the part of the image that contains the object. The cropped image has a fixed size of 200×200 pixels. For SVM training and prediction, it needs to extract features of the objects. The performance of the classification process is greatly

influenced by the selection of features. Selecting suitable features to represent the images as well as suitable kernel and parameters value will increase the performance of classification. In this work, we use an area zoning feature. The object images are divided into 10×10 identical zones. The size of each zone is 20×20 pixels. In each zone, the sum of all white pixels is taken to create a feature. The feature vector with a dimension of 100 is formed from the feature of each zone. Finally, the feature vector is used to train the SVM model for classification. We used Python's scikit-learn to implement SVM training and prediction. Scikit-Learn contains the SVM library, which contains built-in classes for different SVM algorithms.

3. RESULTS AND DISCUSSION

In the experiment, the robot will sort four objects with different shapes as shown in Figure 4. To train an SVM model for classification, we took images of the objects and processed them to obtain a dataset. The dataset consists of 100 images of objects (25 images per object) located at different positions in the workspace. The dataset is then augmented by using the rotation technique. For each cropped image of the object, it is rotated clockwise by a given number of degrees from 0 to 360 to create 72 different images. So, we have obtained a new dataset of 7,200 images. The data augmentation improves the accuracy and generalization of the SVM model. This dataset is split into a training set and a testing set where the percentage of the training set is 80 percent. The features in the training image set are extracted to train an SVM model with a Gaussian kernel. After training, we use the testing set to evaluate the accuracy of the SVM model. The results in Table 1 show the classification accuracy is 99.72%, 99.4%, 99.4% and 99.88% for four objects, respectively.

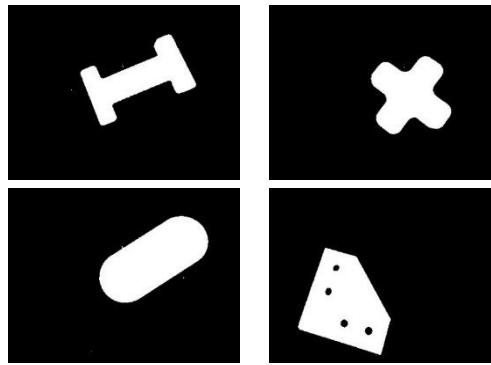


Figure 4. Four objects to classify

Table 1. Classification accuracy

Object	Number testing data	Misclassification	Accuracy
1	360	1	99.72%
2	360	2	99.44%
3	360	2	99.44%
4	360	4	98.88%

To grasp an object and place it in the correct position, we need to determine its 3D coordinate related to the robot base. The 2D image coordinates of the object have been extracted. To transform the 2D image coordinate into the 3D coordinate, it is necessary to know the camera's intrinsic and extrinsic. So, we conduct the camera calibration by using a chessboard and some functions to calibrate the camera in OpenCV. First, we capture 16 images of the chessboard from different viewpoints. To find the corner points in the chessboard, we can use the function, *cv.findChessboardCorners()*. The size of a chess square can be measured by a ruler. So, we have the 3D corner points and image corner points. Then, we can use the function, *cv.calibrateCamera()* to calibrate the camera. This function returns the camera's intrinsic and extrinsic matrix. Using the pinhole model, we can determine the 3D coordinate of the object as follows:

$$\frac{X}{Z} = \frac{u-u_0}{f_x}$$

$$\frac{Y}{Z} = \frac{v-v_0}{f_y}$$

Assuming that the objects have a constant and known height, the camera is mounted on the robot arm to see objects, so the Z values of the objects are known prior. The calibration results give the necessary values are: $f_x = 492.78$; $f_y = 494.30$; $u_0 = 322.17$; $v_0 = 234.55$; $Z = 599.73$ mm. From (8), the X and Y values of the 3D coordinates can be determined. These coordinates are utilized to compute the rotation angles of the joints of the robot by using the inverse kinematic equation of the robot. Finally, the robot is controlled to move from the initial position to the object's position for grasping. Table 2 shows the accuracy when using the proposed algorithm to calculate the 3D coordinate of objects.

Table 2. Accuracy of the 3D estimation method

No.	Image coordinate (pixel)		Estimate value (mm)		Real value (mm)		Error (mm)
	cx	cy	X	Y	X	Y	
1	320	200	147.35	-21.91	145	-24	3.14
2	344	192	176.57	-31.62	180	-31	3.48
3	253	182	65.82	-43.75	63	-40	4.69
4	401	252	245.93	41.17	248	39	2.99
5	422	121	271.49	-117.76	269	-115	3.71
6	325	241	153.44	27.82	152	30	2.61
7	411	155	258.10	-76.51	260	-78	2.41
8	334	167	164.39	-61.95	163	-64	2.47
9	382	258	222.81	48.45	222	51	2.67
10	292	210	113.28	-9.78	111	9	2.41
11	377	270	216.72	63.01	218	62	1.63
12	324	245	152.22	32.67	155	31	3.24

4. CONCLUSION

This paper has trained an SVM to classify four objects with different shapes and a computer vision system is developed to grasp objects thank a 3-DOF robot arm. The dataset consists of 100 images of objects located at different positions in the workspace captured by the camera and processed by the proposed algorithm to extract features for SVM training and calculated the 3D position. The experimental test is performed to verify the possibility of the proposed method. The SVM can classify objects with the number of misclassifications is 9 objects in a total of 1,440 objects.

ACKNOWLEDGEMENTS

We acknowledge Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for supporting this study.




REFERENCES

- [1] V. Cong, "Industrial robot arm controller based on programmable system-on-chip device," *FME Transactions*, vol. 49, no. 4, pp. 1025–1034, 2021, doi: 10.5937/fme2104025C.
- [2] M. Zia Ur Rahman, M. Usman, A. Farea, N. Ahmad, I. Mahmood, and M. Imran, "Vision-based mobile manipulator for handling and transportation of supermarket products," *Mathematical Problems in Engineering*, vol. 2022, pp. 1–10, Jun. 2022, doi: 10.1155/2022/3883845.
- [3] V. Cong, L. Hanh, L. Phuong, and D. Duy, "Design and development of robot arm system for classification and sorting using machine vision," *FME Transactions*, vol. 50, no. 2, pp. 181–181, 2022, doi: 10.5937/fme2201181C.
- [4] H. M. Ali, Y. Hashim, and G. A. Al-Sakkal, "Design and implementation of Arduino based robotic arm," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 12, no. 2, pp. 1411–1418, Apr. 2022, doi: 10.11591/ijece.v12i2.pp1411-1418.
- [5] C.-S. Chen, S.-K. Chen, C.-C. Lai, and C.-T. Lin, "Sequential motion primitives recognition of robotic arm task via human demonstration using hierarchical BiLSTM classifier," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 502–509, Apr. 2021, doi: 10.1109/LRA.2020.3047772.
- [6] P. Sutiyasadi and M. B. Wicaksono, "Joint control of a robotic arm using particle swarm optimization based H2/H ∞ robust control on Arduino," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 2, pp. 1021–1029, Apr. 2020, doi: 10.12928/telkomnika.v18i2.14749.
- [7] A. H. Basori, "End-effector wheeled robotic arm gaming prototype for upper limb coordination control in home-based therapy," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 4, pp. 2080–2086, Aug. 2020, doi: 10.12928/telkomnika.v18i4.3775.
- [8] W.-C. Chang, M.-Y. Cheng, and H.-J. Tsai, "Image feature command generation of contour following tasks for SCARA robots employing image-based visual servoing-A PH-spline approach," *Robotics and Computer-Integrated Manufacturing*, vol. 44, pp. 57–66, Apr. 2017, doi: 10.1016/j.rcim.2016.08.002.
- [9] L. D. Hanh and V. D. Cong, "Implement contour following task of objects with unknown geometric models by using combination of two visual servoing techniques," *International Journal of Computational Vision and Robotics*, vol. 12, no. 5, pp. 464–486, 2022, doi: 10.1504/IJCVR.2022.10048788.
- [10] W. T. Abbood, O. I. Abdullah, and E. A. Khalid, "A real-time automated sorting of robotic vision system based on the interactive design approach," *International Journal on Interactive Design and Manufacturing (IJIDeM)*, vol. 14, no. 1, pp. 201–209, Mar. 2020, doi: 10.1007/s12008-019-00628-w.




- [11] Z. Wang, H. Li, and X. Zhang, "Construction waste recycling robot for nails and screws: computer vision technology and neural network approach," *Automation in Construction*, vol. 97, pp. 220–228, Jan. 2019, doi: 10.1016/j.autcon.2018.11.009.
- [12] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, Sep. 2011, doi: 10.1177/0278364911410755.
- [13] B. Jia, S. Liu, and Y. Liu, "Visual trajectory tracking of industrial manipulator with iterative learning control," *Industrial Robot: An International Journal*, vol. 42, no. 1, pp. 54–63, Jan. 2015, doi: 10.1108/IR-09-2014-0392.
- [14] F. Zha *et al.*, "Semantic 3D reconstruction for robotic manipulators with an eye-in-hand vision system," *Applied Sciences*, vol. 10, no. 3, Feb. 2020, doi: 10.3390/app10031183.
- [15] V. D. Cong and L. D. Hanh, "A new decoupled control law for image-based visual servoing control of robot manipulators," *International Journal of Intelligent Robotics and Applications*, vol. 6, no. 3, pp. 576–585, 2022, doi: 10.1007/s41315-022-00223-5.
- [16] D. Ristic-Durrant, S. M. Grigorescu, A. Graser, Z. Cojbasic, and V. Nikolic, "Robust stereo-vision based 3D object reconstruction for the assistive robot FRIEND," *Advances in Electrical and Computer Engineering*, vol. 11, no. 4, pp. 15–22, 2011, doi: 10.4316/AECE.2011.04003.
- [17] A. Zakhama, L. Charrabi, and K. Jelassi, "Intelligent selective compliance articulated robot arm robot with object recognition in a multi-agent manufacturing system," *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, Mar. 2019, doi: 10.1177/1729881419841145.
- [18] H. Wei, X.-X. Chen, and X.-Y. Miao, "Vision-guided fine-operation of robot and its application in eight-puzzle game," *International Journal of Intelligent Robotics and Applications*, vol. 5, no. 4, pp. 576–589, Dec. 2021, doi: 10.1007/s41315-021-00186-z.
- [19] O. Abdullah, W. Abbood, and H. Hussein, "Development of automated liquid filling system based on the interactive design approach," *FME Transactions*, vol. 48, no. 4, pp. 938–945, 2020, doi: 10.5937/fme2004938A.
- [20] M. Intisar, M. M. Khan, M. R. Islam, and M. Masud, "Computer vision based robotic arm controlled using interactive GUI," *Intelligent Automation & Soft Computing*, vol. 27, no. 2, pp. 533–550, 2021, doi: 10.32604/iasc.2021.015482.
- [21] N. Goyal, K. Gupta, and N. Kumar, "Clustering-based hierarchical framework for multiclass classification of leaf images," *IEEE Transactions on Industry Applications*, vol. 58, no. 3, pp. 4076–4085, May 2022, doi: 10.1109/TIA.2022.3153757.
- [22] T. J. Barry and C. R. Nagarajah, "Object recognition in industrial environments using support vector machines and artificial neural networks," *The International Journal of Advanced Manufacturing Technology*, vol. 48, no. 5–8, pp. 815–821, May 2010, doi: 10.1007/s00170-009-2313-3.
- [23] P. M. Dinesh and R. S. Sabenian, "Comparative analysis of zoning approaches for recognition of Indo Aryan language using SVM classifier," *Cluster Computing*, vol. 22, no. S5, pp. 10955–10962, Sep. 2019, doi: 10.1007/s10586-017-1258-9.
- [24] V. Kurilová, J. Goga, M. Oravec, J. Pavlovičová, and S. Kajan, "Support vector machine and deep-learning object detection for localisation of hard exudates," *Scientific Reports*, vol. 11, no. 1, Aug. 2021, doi: 10.1038/s41598-021-95519-0.
- [25] A. Alqudah and A. M. Alqudah, "Sliding window based support vector machine system for classification of breast cancer using histopathological microscopic images," *IETE Journal of Research*, vol. 68, no. 1, pp. 59–67, 2022, doi: 10.1080/03772063.2019.1583610.
- [26] K. Sashi Rekha and S. A. Miruna Joe Amali, "Efficient feature subset selection and classification using levy flight-based cuckoo search optimization with parallel support vector machine for the breast cancer data," *International Journal of Imaging Systems and Technology*, vol. 32, no. 3, pp. 869–881, May 2022, doi: 10.1002/ima.22686.
- [27] E. Byvatov, U. Fechner, J. Sadowski, and G. Schneider, "Comparison of support vector machine and artificial neural network systems for drug/non-drug classification," *Journal of Chemical Information and Computer Sciences*, vol. 43, no. 6, pp. 1882–1889, Nov. 2003, doi: 10.1021/ci0341161.
- [28] K. Crammer and Y. Singer, "On the algorithmic implementation of multiclass kernel-based vector machines," *Journal of Machine Learning Research*, vol. 2, pp. 265–292, 2001.
- [29] Chih-Wei Hsu and Chih-Jen Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002, doi: 10.1109/72.991427.

BIOGRAPHIES OF AUTHORS



Vo Duy Cong    received the M.Sc. degree in Mechatronics Engineering from Ho Chi Minh City University of Technology in 2021. He joined the Industrial Maintenance Training Center, Ho Chi Minh City University of Technology, Vietnam National University-Ho Chi Minh City, since 2018. His research interests are mechanical design, LQR control, LQG control, model predictive control, industrial robot, mobile robot, visual servoing control, system modeling, computer vision, embedded system programming, automation system, industrial maintenance. He can be contacted at congvd@hcmut.edu.vn.



Thai Thanh Hiep    Thai Thanh Hiep received M.Sc. degree in Mechatronics Engineering from Ho Chi Minh City University of Technology in 2022. He joined the department of Automotive Engineering, Faculty of Transportation Engineering, Ho Chi Minh City University of Technology, Vietnam National University Ho Chi Minh City, since 2021. His research interests are special vehicles, hydraulic and pneumatic systems, automatic and semi-automatic control systems, internal combustion engines, simscape multibody, and SimHydraulics. He can be contacted at thaithanhhiiep@hcmut.edu.vn.