Implementation of Convolutional Neural Network (CNN) Algorithm for Autonomous Robot

Tiara Kusuma Dewi, Kushendarsyah Saptaji, Adven Simarmata*
Department of Mechanical Engineering,
Faculty of Engineering and Technology, Sampoerna University,
Jakarta 12780, INDONESIA
**kushendarsyah@sampoernauniversity.ac.id*

Muhamad Rausyan Fikri
Automation Technology and Mechanical Engineering,
Faculty of Engineering and Natural Science, Tampere University,
Tampere 33720, FINLAND

ABSTRACT

In Indonesia, the development of autonomous robots has emerged intensively since the last coronavirus pandemic, especially the autonomous UV disinfection (A-UV) robot. A-UV disinfection robot has the purpose of purifying germs and pathogens in critical areas, such as the hospital. As the minuscule creature can be difficult to control, the anticipation of letting no human have contact with it is one of the other purposes of the A-UV disinfection robot. However, the systematic development of the autonomous robot is the priority, where the robot can offer a collision-free obstacle, and target-lock when arriving at the designated location. In this study, two main contributions are proposed to develop the autonomous robot: 1) Convolutional Neural Network (CNN) algorithm to learn the potential surrounding the lock area from the dataset to ensure collision-free during the operation. 2) Original design to ensure the compactness of the autonomous robot with almost omnidirectional UV light. We design the surrounding area with "BOX" as the obstacle and "SIGN STOP" as the target in our CNN dataset. The performance is validated to have 97% and 99% for training and validation performance and 0.3% for loss. The robot prototype was also developed and tested inside a workspace with a size of 2.1 x 3 m. The robot prototype successfully performed the required tasks.

ISSN 1823-5514, eISSN 2550-164X

https://doi.org/10.24191/jmeche.v21i2.26264

Received for review: 2023-07-12
Accepted for publication: 2024-02-29
Published: 2024-04-15

Keywords: *Autonomous Robot; Convolutional Neural Network Application, Object Classification, Vision Based - A-UV Disinfection Robot; Mobility Testing*

Introduction

The key to the autonomous robot is to perform its task with little or no human intervention. Autonomous robots require no supervision; they can work side-by-side or even substitute humans to perform tasks that humans cannot, should not, or do not want to do, such as in hazardous and indicated fast pathogens distribution areas. Autonomous robots have been applied in many industries and applications, such as manufacturing [1], agriculture [2], warehouse management systems [3]-[4], medical [5], and military applications [6]. In addition, the autonomous robot can also be implemented in exploration [7], home services [8], surveillance [9], and potentially for disinfection purposes [10]-[13].

Despite various applications of autonomous robots, the autonomous robot requires a range of abilities to perform its designed task effectively. One of them is to understand their surrounding and avoid obstacles by either using sensor perception or object recognition. In sensor perception, deep learning such as K-Nearest Neighbours (KNN) and Artificial Neural Networks (ANN) is commonly used to process the data from the sensor and use it to help the robot sense its working area.

KNN is a simple supervised machine learning algorithm that can solve classification problems. It classifies new data based on the distance of the new data to the nearest data or its neighbour in terms of characteristics. In 2018, Dairi et.al used KNN to detect the presence of obstacles in intelligent transportation systems applications [14]. Gao et al. [15], used KNN for the path planning tasks in space robotic manipulators. Meanwhile, Asti et al. [16] implemented the KNN algorithm to train a quadcopter to make efficient avoidance decisions. The KNN algorithm has been used to increase the accuracy of autonomous robot localization, particularly in indoor environments [17].

In contrast, ANN is a set of mathematical and algorithmic methods for solving many problems especially related to processing sensor data. It is a simplified model of a biological neuron that receives a signal and subsequently processes it. In the obstacle avoidance application, ANN receives a signal from the sensor in the form of data and then processes it to decide the torque needed to avoid the obstacle. In 2018. Zakaria et al. [18] used the ANN algorithm to process the data from multiple sensors such as a potentiometer and inertia motor to recognize the changing behaviour in the lane for collision avoidance purposes. Farias et al. [19], Sochima et al. [20], and Andreev and Tarasova [21], used the ANN algorithm to design motion control systems in mobile robots, mainly to avoid obstacle decisions from processing the ultrasonic sensor data. Medina et al. [22] and Farag et al. [23] applied the ANN algorithm on a mobile robot to avoid collision while controlling movements to follow the pre-defined path. Besides, the application on mobile robots for obstacle avoidance using the ANN has been applied in [24], where the proposed ANN control system is used to help the robot perform local navigation. In another case, ANN was applied to control the robot's velocities to be able to track the trajectories [25]. ANN combined with robust feedback control is successfully applied to improve control performances and to increase the robust stability of the closed-loop control system in the Two Wheel Mobile Robot (TWMR) [26]. The system output result shows the error between the neuro controller and the original controller in terms of both linear and angular velocity is significantly small 10^{-7} resulting in the ability of TWMR to track the reference input accurately.

Besides using sensor perception, autonomous robots also can sense their surroundings using the ability of object recognition. The most popular deep learning algorithm for object recognition is the Convolutional Neural Network (CNN). It provides a potential capability in obstacle avoidance applications by using object detection ability to ensure collision-free during the operation of autonomous robots. Instead of processing data from sensors, CNN, a classification-machine algorithm, can be used to train a vision-based model of obstacle avoidance using a real-time captured image from the camera. The CNN algorithm has been used for obstacle avoidance applications in many devices. Mechal et al. [27] applied CNN using RGB Depth Image Fusion for obstacle avoidance applications in robots. Duan et al. [28] have applied the same algorithm to experiment with real-time computer vision for obstacle detection in automatic lawnmower applications. Ma et al. [29] used the CNN algorithm to perform obstacle detection in the unmanned surface vehicle (USV), and Zhang et al. [30] used CNN for depth estimation in the unmanned aerial vehicle (UAV) application. Chang et al. [31] used the CNN algorithm for the image-based obstacle avoidance method with a monocular webcam in mobile robot navigation. Che et al. [32] used a stereo camera combined with OpenCV for object tracking. Osman et al. [33] use a deep-CNN algorithm to detect pavement cracks. Furthermore, several machine learning models that use CNN architecture in their pipeline that are ready to be used for autonomous vehicle applications, namely YOLO [34]-[35], Region-CNN (R-CNN) [36], Mask R-CNN [37], MobileNet [38], and SqueezeNet [39]. Thus, CNN has advanced object detection and classification especially when the dataset and environment are known, such as in routine critical areas like hospitals.

The implementation of obstacle avoidance using KNN and ANN methods mostly used sensors as the main input for the algorithm. The use of sensors has limitations and is inadequate in identifying the object size and the distance range between the sensor and the objects in the surroundings. These limitations can be complicated in the application of complex environments.

Thus, an alternative method deserves to be investigated. One of the alternative methods is the implementation of vision-based obstacle avoidance using CNN. This method is expected to overcome the limitations of KNN and ANN especially in identifying the object size and shape. This deep learning method is known to be simpler because it needs only a camera instead of sophisticated equipment, such as light detection and ranging sensors (LIDARs) or Robo Peak Lidar (RPLidar) to track and classify an object [32].

One of the developments of autonomous robots using the CNN algorithm is in the disinfection application. The use of autonomous robots using UV for disinfection purposes has been significantly increasing worldwide, especially in Indonesia since the last coronavirus pandemic. In Indonesia, several disinfection robots have been reported and developed by some researchers, such as an Autonomous UVC Mobile Robot (AUMR) [37] and UV-based COVID-19 Sterilization Robot VIOLETA [38]. AUMR robot can be operated using autonomous mode by using a line tracking system combined with an ultrasound-based obstacle avoidance system. Besides, the VIOLETA robot is developed using a teleoperation system when the UV robot operates manually by the operator using the wireless remote. However, the exploration of object detection-based obstacle avoidance autonomous robots has not been that popular yet.

Therefore, the objective of this paper is to implement the CNN algorithm in the autonomous robot for ultraviolet (A-UV) disinfection application. CNN algorithm is implemented to give the robot the ability to detect objects in its path and avoid obstacles. This paper consists of three main parts; (i) the material and methods part that discusses the development of the CNN algorithm and A-UV disinfection robot design, (ii) the result and discussion part that discusses the performance of the CNN algorithm in A-UV disinfection robot application, and (iii) concludes the finding and contribution of this research.

Materials and Methods

The methodology to produce the autonomous robot for UV (A-UV) disinfection can be divided into three stages, (a) development of an A-UV robot prototype, (b) development of a machine learning model using the CNN algorithm, and (c) development obstacle avoidance algorithm.

Development of A-UV robot prototype

There are two steps to build the A-UV disinfection robot such as designing the robot and building the prototype. A similar design flow is used in this research [40].

The designing process consists of two parts. The first is to design the robot's chassis and its components, and the second is to design the electrical assembly. A 3D design software is used to design the robot's chassis and its components. All robot chassis parts are designed manually, and some components inside the robot are obtained from open source.

Three design requirements are needed to be fulfilled. Firstly, the A-UV disinfection robot should be implemented in a small room of 2.4 × 3 in m. Secondly, the UVC wavelength should be well distributed to get the maximum benefits. Any objects should not be blocked by the radiation of UVC wavelength. Thirdly, the robot can move autonomously to limit human contact in its operation.

The robot dimension of 300 ×300 × 675 mm is proposed so that the design's requirements can be achieved. The robot is divided into two main parts i.e., the main box and the UVC lamp's stand. The main box itself is divided into two parts which are the lower part for the power supply, and the upper part as the electrical circuit box. The lamp's stand consists of a single cantilever where a maximum of 4 UVC lamps can be installed. It is specially designed for a 15 W UVC tubular lamp with a height of 450 mm. Then, a machine-learning model is implemented on the robot. The robot is trained using CNN and obstacle avoidance algorithm so it can search the targeted location and avoid the obstacle fully autonomous. Table 1 gives a summary of the design's requirements and the proposed design.

Table 1: Design's requirement and proposed design for A-UV robot development

Requirement	Proposed
• Can be implemented in a small patient room of 2.4 × 3 m and has a maximum height of 900 mm.	• The chassis' dimension is 300 × 300 × 675 mm for length, width, and height.
• UVC wavelength should be well distributed.	• Single cantilever that can hold max 4 UVC lamps.
• Can move autonomously.	• ML Implementation.

Once the robot's chassis is designed, and its components are arranged, its electrical components are assembled. The electronic part of the A-UV robot consists of the perception, cognition, actuation, power supply system, and UVC lamp controller. Perception is a system that gives the robot the ability to learn about its surrounding environment. It consists of a USB camera to detect objects and an ultrasonic sensor to measure distance. Cognition is known as the decision-maker system. It is responsible for collecting and processing the input data and then deciding the corresponding action regarding its movement. The cognition system used in this robot is Raspberry Pi 4B. An actuation system is to perform the movement. It includes a 12 V motor driver and wheel. Lastly, a wet battery of 5 V and an inverter are used as the power supply, and an additional electronic component is a 5 V relay and an electronic ballast to control the UVC lamps during the disinfection process. Table 2 gives a summary of the UV robot's electrical components.

Table 2: Design's requirement and proposed design for A-UV robot development

System	Electrical components
Perception	USB camera Logitech c525 Ultrasonic sensor HC-SR04
Cognition	Raspberry Pi 4B
Actuation	12V DC motors Controlled wheels Motor driver L298N
Power supply	Wet battery 12V DC and 5 Amps/Hr Inverter (12V DC to 220 AC)
UVC lamp controller	Electronic ballasts 5V Relay

After designing the chassis and electrical diagram, the prototype is built. The robot's chassis is made of aluminium alloy materials because of their lightweight and non-corrosive behaviour [41]. An aluminium rectangular hollow tube of 30 × 40 × 1 mm for length, width, and thickness, respectively, is used as the main chassis structure. During the prototyping process, self-riveting piercing (SRP) is used to join parts. SRP includes a forging process that uses a rivet to join parts. The rivet itself is a permanent mechanical fastener, where accuracy is needed during the prototyping process.

Developing a machine learning model using CNN algorithm

After the prototype is built and electrical components are attached, the robot is trained to perform image classification using the Convolutional Neural Network (CNN) algorithm. There are two classes trained in this process. First is the image of a "BOX," which later be classified as an obstacle. Second, is the image of a "STOP SIGN" classified as a target. The machine learning framework consists of several steps, including loading the dataset, pre-processing the data, building the machine learning modelling using a convolutional neural network (CNN) algorithm, and evaluating and tuning the model. During the evaluation process, the model experiences overfitting. Hence remodelling the CNN algorithm is performed to respond to the situation. Splitting the data and drop-out regularization are selected to solve the problems. Once the model fits the assigned task, the CNN-based machine-learning model can be deployed to the robot. Figure 1 shows the framework of machine learning used in this paper.

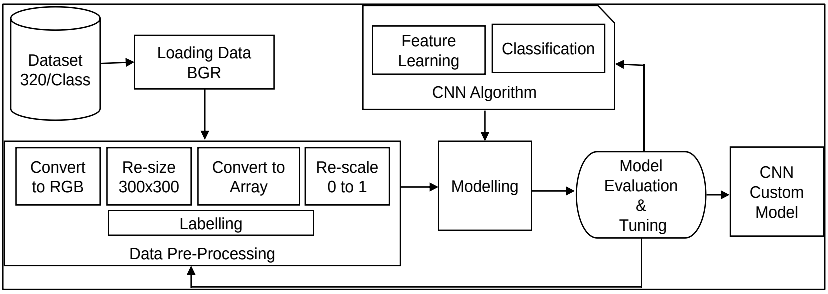


Figure 1: Machine learning framework

This machine learning model is trained using 40 iterations, and its performance is evaluated based on its precision, recall, and f1-score. Once the classifier model is optimized, the model is ready to be deployed to the A- UV robot. The A-UV robot utilizes the pre-trained classifier model to classify objects in its environment in real time.

Real-time image classification on the Raspberry Pi 4B, the main controller of the A-UV robot, is achieved using the TensorFlow Lite Interpreter and OpenCV (Open-Source Computer Vision Library). Initially, the custom CNN model is converted into TensorFlow Lite file type. Along with the class labels, the converted model is then uploaded to the Raspberry Pi. With both files in place, the A-UV robot is prepared for image classification.

The live webcam feed captures the input in video form, and OpenCV receives and processes the images from the webcam. Then, the images are sent to the classifier model to learn and classify the features of the captured images in real time. The resulting classified image is provided to the robot, allowing it to execute tasks based on the classification output.

If the model classifies the detected object as a "BOX," the A-UV robot interprets it as an obstacle to be avoided. The robot executes a left turn according to the programmed instructions to navigate around the obstacle. Conversely, if the identified object is a "STOP SIGN," the robot recognizes it as a target to approach. The flow chart in Figure 2 outlines the assigned tasks.

As shown in Figure 2, the A-UV robot is assigned to perform real-time image classification and distance measurement. Depending on the classification result, the robot will either autonomously avoid or approach the detected object. To assess its mobility performance, the A-UV robot undergoes testing in a 2.1 × 3 m workspace with arrangements of "BOX" and "STOP SIGN" objects.

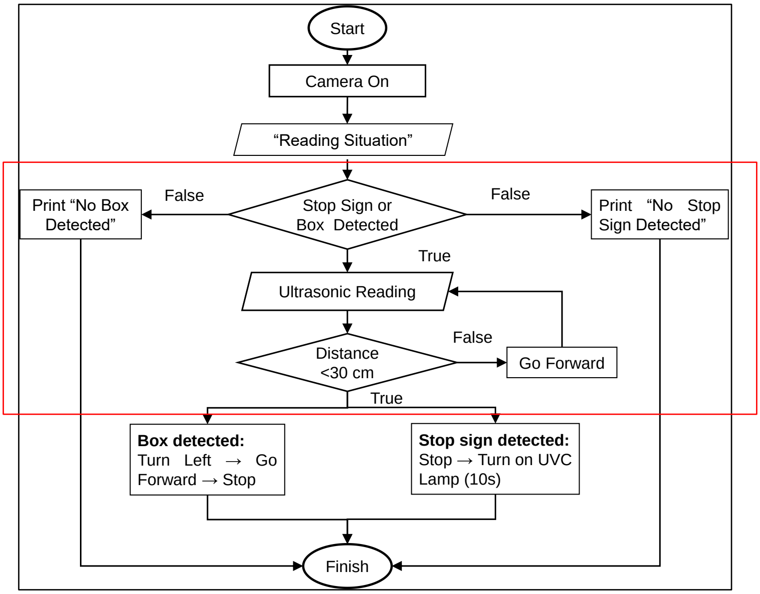


Figure 2: Flowchart of the algorithm of assigned task for "BOX" detection and "STOP SIGN" detection

Developing obstacle avoidance algorithm

The robot is trained with a machine learning model to know the difference between the BOX and STOP as the surrounding obstacle and target, respectively. In our algorithm, (see Figure 2), the red colour rectangular shape indicates the specific collision-free that we need to introduce in our systems. Let us assume that the area of the autonomous robot will travel is C_shape consisting of C_free and C_obs. Now that C_shape has been presented and consider the parts of C that are prohibited to travel to ensure collision-free, (see Figure 3).

Let $A(q) \subset$ W denotes a closed set of points in the world occupied by the A-UV robot A with configuration q states. A configuration $q \in C$ places the robot into a collision if and only if $A(q) \cap O = \emptyset$. (Here, the robot and obstacle are attempting to occupy at least one common point in W). We follow [42] to determine the set of all non-colliding configurations or free space which is defined in Equation (1) as:

$$\mathcal{C}_{\text{free}} = \{q \in \mathcal{C} | \mathcal{A}(q) \cap \mathcal{O} = \emptyset \tag{1}$$

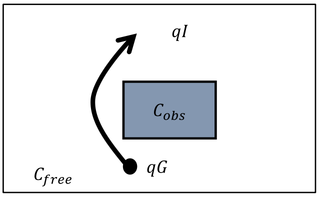


Figure 3: In the C_space the problem looks simply: Connect q_I to q_G while remaining in C_free

As shown in Figure 3, the working space for the experiment will contain one box as the representation of the obstacle. The camera will detect the existence of the box while the ultrasonic sensor keeps measuring the distance. Once the robot has reached its maximum distance from the obstacle, the obstacle avoidance algorithm is activated to guide the robot in navigating around the obstacle.

Results and Discussion

The result and discussion are divided into three stages, (a) developing rA-UV robot prototype, (b) testing the performance of the CNN algorithm in classifying objects, and (c) testing the mobility of the A-UV disinfection robot.

Development of A-UV robot prototype

The proposed A-UV robot structure consists of 3 levels. They are lamps stand, electronic box, and power supply box. The lamp's stand is built with a height of 450 mm since it is designed for a 15-W UVC lamp. The electronics box has dimensions of 180 × 300 × 90 mm, and the power supply box is 300 × 300 × 135 mm for width, length, and height, respectively. Therefore, the total dimension of the robot is 300 × 300 × 675 mm. Figure 4 shows the robot's dimensions along with its part's name.

The lamp's stand consists of two 15 W UVC lamps and a fitting tabular lamp. It is made from a rectangular aluminium hollow tube. All wiring connecting the lamp to the power supply is located within the tube to protect it from radiation. The electrical circuit box consists of Logitech c525, ultrasonic sensor HC-SR04 and breadboard, motor driver l298n, Raspberry Pi 4B, and an additional small brown electronic box. The small electronic box is used to secure the additional wiring diagram. Meanwhile, the power supply box consists of two electronic ballasts, an inverter, and a wet battery. In addition, the bottom part of the A-UV robot consists of two controlled wheels along with its motor encoders and stand, and a single rolling caster wheel at the front side.

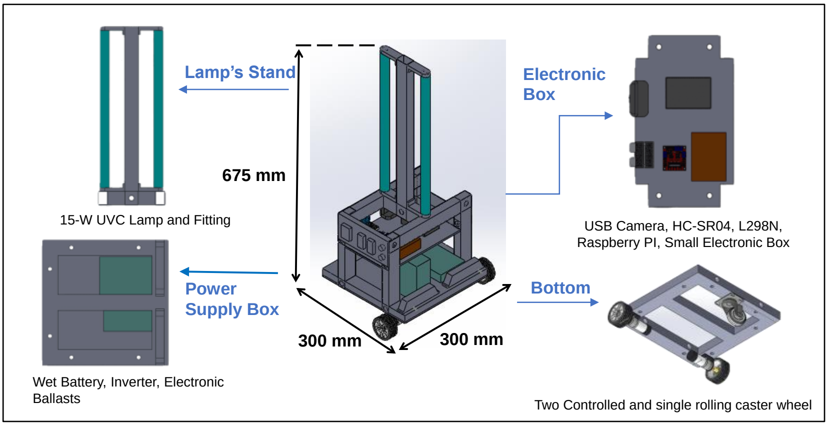


Figure 4: A-UV robot's dimension and component arrangement

The electrical assembly is divided into two parts. First, the electrical assembly of the power supply and UVC lamp controller. As shown in Figure 5, the electrical diagram connects the wet battery and inverter to supply the power needed for the motor driver (12 V DC), UVC lamp (220 V), and Raspberry adaptor (220 V). The 12 V and 5 V- relays are attached to the electrical schematic to prevent signal interference, enabling the Raspberry Pi to control both the motor driver and the UVC lamp. In addition, a switch and indicator lamp were also installed to provide visual feedback on the operational status of the A- UV disinfection robot.

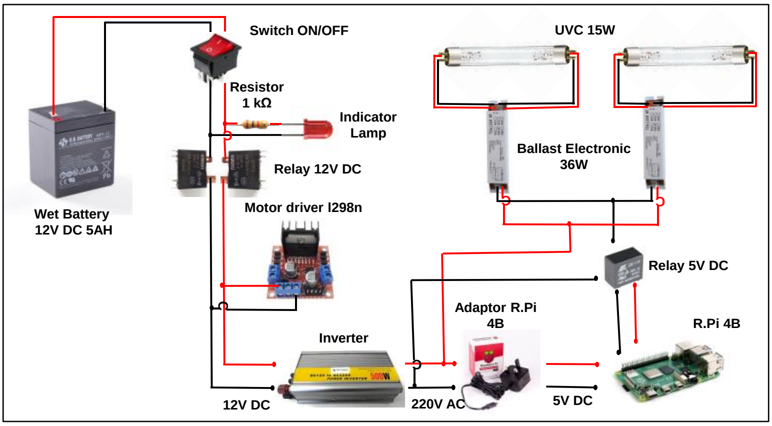


Figure 5: Schematic of power distribution

Second, the electrical assembly that connects the perception and actuation system to the cognition system. Figure 6 shows the electrical diagram, between Raspberry Pi 4B, motor DC, ultrasonic sensor, USB camera, 5 V relay, and UVC lamps.

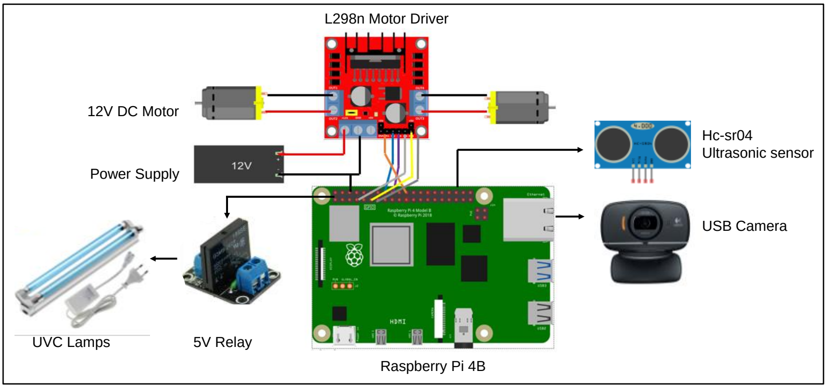


Figure 6: Electrical diagram of cognition, perception, and actuation system

The proposed structural building and electronic assembly are then assembled as a complete A-UV robot prototype. Figure 7 shows the A-UV robot prototype, along with its components attached and arranged.

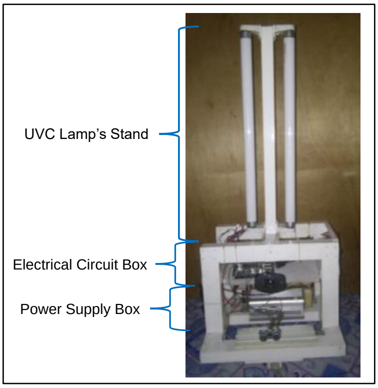


Figure 7: Prototype of A-UV disinfection robot

The prototype refers to the proposed design with dimensions of 300 × 300 × 675 mm and consists of three main parts which are UVC Lamp's stand, electrical circuit box, and power supply box. It is made of aluminium alloys materials because of their lightweight and non-corrosive behaviour [41].

Performance of machine learning model using CNN algorithm

A machine learning model with the CNN Algorithm is developed to train the robot to understand its surroundings and avoid obstacles. The architecture of CNN used in this paper is shown in Figure 8. The architecture of CNN used to train the autonomous UV disinfection robot consists of an input image, feature learning, classification process, and output in the form of the image's label. The input of the CNN model are images of "BOX" and "STOP SIGN". These images are then fed to the CNN model consisting of two main processes which are feature learning and classification process. Then, the output of the CNN model is in the form of the classified images' labels such as "BOX" and "STOP SIGN". Each process of CNN is described below.

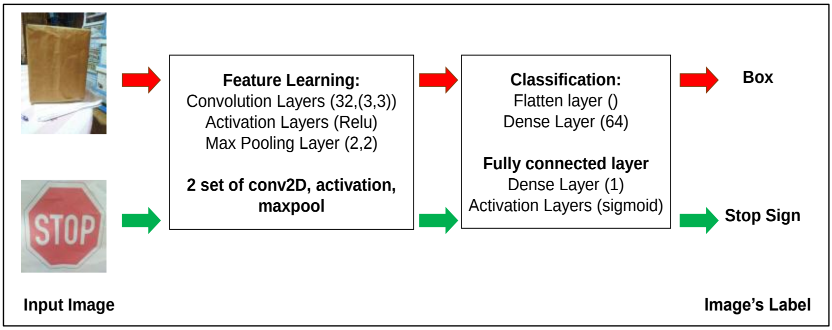
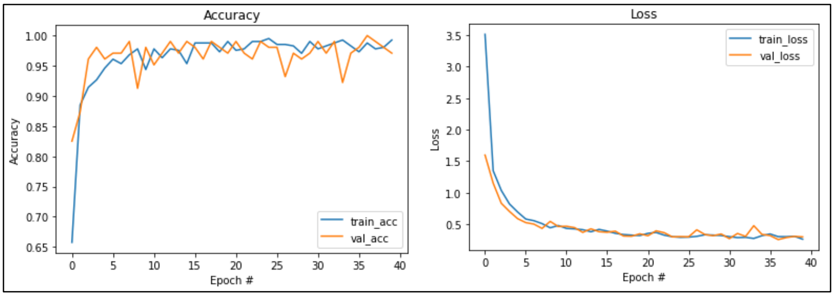


Figure 8: Architecture of CNN model for image classification

Feature learning consists of two layers which are the convolution and pooling layers. The convolution is used to convert an image using the convolution function of some algorithm operation. It is a weight matrix that acts as a digital filter to extract the high-level features such as edges, colour, and gradient orientation from the input image [43]. Each filter produces a different output. In this paper, the Conv 2D layer is used due to the form of an image for the CNN input. The pooling layers are employed to combine the neighbouring pixels into a single pixel. There are two types of pooling layers which are the mean pooling layer and max-pooling layer [43]. In this paper, the max-pooling layer is implemented because it has good performance in pixel reduction and low-feature extraction. During the feature learning process, the Rectified Linear Unit (ReLu) activation layer is used. ReLu activation function is preferable because it can shorten the feature extraction process by randomly activating some neurons.

The classification consists of flattened, fully connected, and activation layers. The feature map generated from the feature learning and extraction process is in the form of a multidimensional array. It needs to "flatten" or shape into a vector to fill the requirement as an input for the fully connected layer. In the fully connected layer, all activity neurons from the previous layer are connected to neurons in the next layer. The fully connected layer is usually used in the deep learning method to process the data to be classified. Since there are only two classified objects ("BOX" and "STOP SIGN"), the sigmoid activation layer is used in this machine-learning architecture. To assess the performance of the CNN image classification model, the difference in validation and training accuracy and loss from each iteration are observed and shown in Figure 9.

The training and validation accuracy of the model after going through 40 iterations reached about 97% and 99%, respectively (Figure 9a). The loss can be minimized down to about 0.3% (Figure 9b). The model performance in classifying the images cannot be assessed only by observing the training graph. Hence the model's precision, recall, and f1-score should be calculated based on the true-false model in classifying the images [33]. The performance of the CNN model in classifying the image can be seen in Figure 10.



(a) (b)

Figure 9: Accuracy (left) and loss (right) between training and validation performance of custom CNN model in the system

As shown in Figure 10, from the total of 67 images of "BOX" being tested, the model can classify 65 of them correctly. Meanwhile, from the total of 61 images of "STOP SIGN", the model successfully classified and labelled 58 of them correctly. The performance of the model can be evaluated and calculated using the following Equations (2)-(4) [33]:

$$Precision = TP \div (TP + FP) \tag{2}$$

$$Recall = TP \div (TP + FN) \tag{3}$$

$$F1 = \frac{(2\times Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

where TP is True Positive, FP is False Positive, and FN is False Negative.

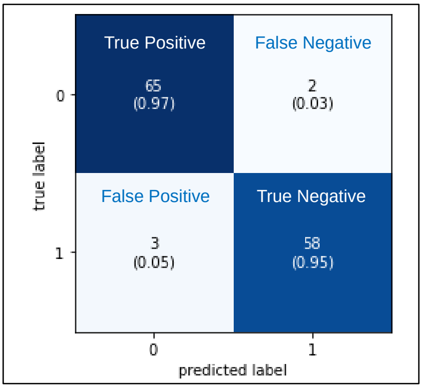


Figure 10: Predicted label (x-axis) actual label (y-axis)

Using the information in Figure 10 and Equations (2), (3), and (4), the performance of the CNN model in performing image classification can be evaluated. Table 3 shows the report of CNN model performance.

Table 3: Classification performance report

Label	Precision	Recall	F1-score
(Box)	0.96	0.97	0.96
(Stop sign)	0.97	0.95	0.96

Based on the evaluation process, the CNN model has performed a good image classification task. Hence, this model is deployed to Raspberry Pi to help the A-UV robot detect the object in its path in real time. Figure 11 shows the output of CNN model in Raspberry Pi. As shown in Figure 11, the CNN model successfully classified the captured image and labelled it accordingly.

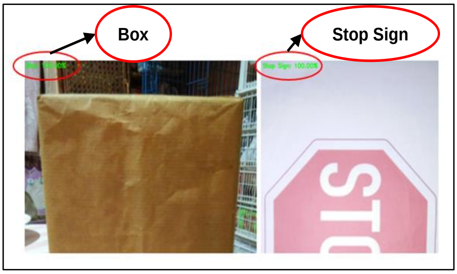


Figure 11: Predicted label (x-axis) actual label (y-axis)

Real-time image classification is achieved using the TensorFlow Lite and OpenCV. Firstly, the pre-trained model is converted into TensorFlow Lite. Along with the class labels, the pre-trained model is then uploaded to the Raspberry Pi. Webcam and OpenCV are used to capture the images in real-time and then feed them to the pre-trained model as the input. The performance of real-time image classification itself can reach up to 30 frames per second (FPS). However, its performance is decreasing due to the combination with other tasks like obstacle avoidance and disinfection process.

Performance of A-UV disinfection robot using CNN algorithm

After the prototype is built and the machine learning model is trained, the machine learning model is then deployed to the robot's cognition system. The experiment was then performed by assigning a task to the robot. The robot operated its task in the indoor workspace with the size of 2.1 x 3 m with an ambient light environment. Figure 12 shows the performance of the robot in trajectory movement. Step (1), the robot moves straightaway. Once an obstacle in the form of "BOX" was detected in front of it with a distance less than 30 cm, it was then avoided by turning to the left (Step (2)). It then moves back to its original trajectory path (Step 3) until the "STOP SIGN" is detected. The robot approached it until the distance was less than 30 cm. The robot stopped and performed the disinfection process by automatically switching on UVC lights for about 10 seconds (Step 4). Based on this experiment, the robot has performed its assigned task successfully.

As shown in Figure 12, the A-UV disinfection robot can perform its disinfection task autonomously without any human interference. In addition, this result proves that the proposed autonomous A-UV robot using the CNN method can be an option for A-UV disinfection.

Conclusion

In this paper, an autonomous robot prototype for A-UV disinfection application has been successfully developed using the CNN algorithm. The CNN algorithm is used to provide ability to the robot. Two main contributions of our study have been delivered and achieved successfully. Firstly, the CNN algorithm model has been developed with training and validation accuracy of 97% and 99%, respectively, after 40 iterations. Secondly, the compact design of the A-UV is achieved with the dimension of the proposed prototype to fit to narrow workspace area has been fulfilled. The prototype robot with a dimension of 300 ×300 ×675 mm is built consisting of a power supply box and electrical circuit box at the lower part, and a UVC lamp's stand at the upper part. It is equipped with a unique lamp stand that allows the UVC lamp to radiate its wavelength uniformly. As a result, the autonomous UV robot can perform the task autonomously based on the output of the classified captured object during the trajectory movement testing. The autonomous robot can avoid and detect obstacles represented by a box and stop sign, and approach the aim successfully. These results imply that the proposed autonomous UV disinfection robot using the CNN algorithm is reliable and can be applied in deep learning vision-based applications. For future work, we aim to formalize the collision-free of the autonomous robot to have a safety guarantee operation, where the mathematical definition will be delivered and incorporated with the CNN to enhance the performance. Furthermore, the integration of Simultaneous Localization and Mapping (SLAM) technology and Robot Operating System (ROS) will be explored. SLAM technology will facilitate the robot to map its working environment. Subsequently, the map that is created will be used by the ROS to tell the robot its current location within the environment. The combination of CNN image classifier, obstacle avoidance, SLAM, and ROS will ensure robot capability for autonomous operation in a real application environment such as a hospital.

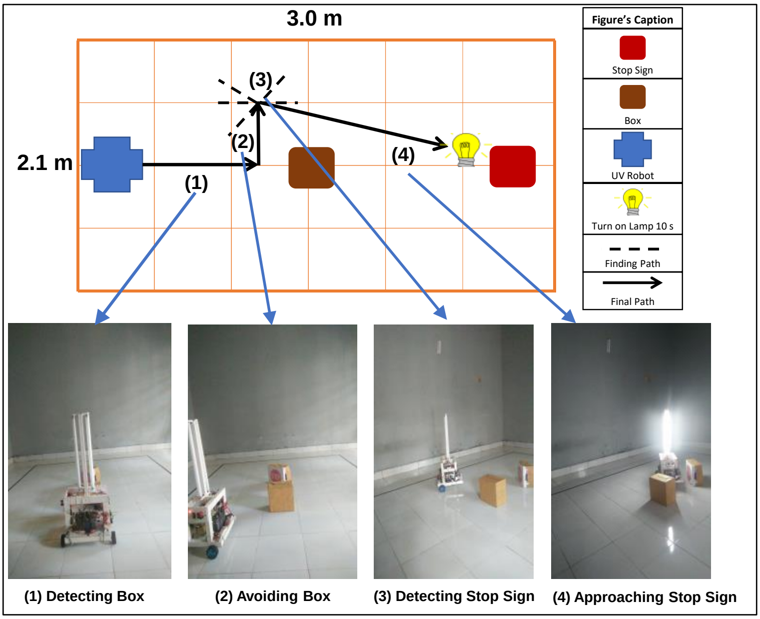


Figure 12: A-UV robot (above) trajectory movement from top view (below) performs the task autonomously

Contributions of Authors

The authors confirm the equal contribution in each part of this work. All authors reviewed and approved the final version of this work.

Funding

This research was supported by internal research grant from the Center of Research and Community Service (CRCS) Sampoerna University and *Koperasi Siswa Jasa Bangsa* (KJSB).

Conflict of Interests

All authors declare that they have no conflicts of interest.

Acknowledgment

The authors would like to thank Internet of Things (IoT) Laboratory and Mechanical and Integrated Manufacturing (MIM) Laboratory Sampoerna University.

References

[1] F. Perez-Grau, J. R. M. Dios, J. L. Paneque, J. J. Scevedo. A. T. Gonzalez, A. Viguria, J. R. Astorga, and A. Oilero, “Introducing autonomous aerial robots in industrial manufacturing,” *Journal of Manufacturing Systems*, vol. 60, pp. 312–324, 2021. doi: 10.1016/j.jmsy.2021.06.008

[2] A. A. Kulkarni, P. Dhanush, B. Chetan, C. Gowda, and D.-P. Shrivastava, “Applications of automation and robotics in agriculture industries; A review,” *IOP Conference Series Materials Science and Enginerring*, vol. 748, pp. 1–8, 2020. doi: 10.1088/1757-899X/748/1/012002

[3] J. Bačík, P. Tkac, L. Hric, S. Alexovic, K. Kyslan, R. Olexa, and D. Perdukova, “Phollower—The universal autonomous mobile robot for industry and civil environments with COVID-19 germicide addon meeting safety requirements”, *Applied Sciences*, vol. 10, no. 21, pp. 1–16, 2020. doi: 10.3390/app10217682

[4] Mohd. Nayab Zafar, J. C. Mohanta, and A. Sanyal, “Design and implementation of an autonomous robot manipulator for pick & place planning,” *IOP Conference Series Materials Science and*

Engineering, vol. 691, no. 1, pp. 1–9, 2019. doi: 10.1088/1757-899X/691/1/012008

[5] P. Pinto, A. Varghese, F. Talari, S. Edward, and A. Pabarekar, "Autonomous hospital assistant bot," *ITM Web of Conferences*, vol. 44, pp. 1–6, 2022. doi: 10.1051/itmconf/20224403047

[6] D. Tendulkar, M. Ansari, D. Patil, R. Bhatlekar, V. Pawar, and S. Aswale, "Design and implementation of autonomous robot for military applications using artificial intelligence for performance management," *International Journal of Engineering Research and Applications*, vol. 10, no. 9, pp. 13–24, 2020. doi: 10.9790/9622-1009011324

[7] S. Liu, S. Li, L. Pang, J. Hu, H. Chen, and X. Zhang, "Autonomous exploration and map construction of a mobile robot based on the TGHM algorithm", *Sensors*, vol. 20, no. 2, p. 490, 2020. doi: 10.3390/s20020490

[8] S. Y. Tan and A. Taeihagh, "Governing the adoption of robotics and autonomous systems in long-term care in Singapore", *Policy and Society*, vol. 40, no. 2, pp. 211–231, 2021. doi: 10.1080/14494035.2020.1782627

[9] M. Diwakar, K. Sharma, R. Dhaundiyal, S. Bawane, K. Joshi, and P. Singh, "A review on autonomous remote security and mobile surveillance using internet of things," *Journal of Physics: Conference Series*, vol. 1854, no. 1, pp. 1–6, 2021. doi: 10.1088/1742-6596/1854/1/012034

[10] K. Cresswell and A. Sheikh, "Can disinfection robots reduce the risk of transmission of sars-cov-2 in health care and educational settings?," *Journal of Medical Internet Research*, vol. 22, no. 9, pp. 1–3, 2020. doi: 10.2196/20896

[11] F. Astrid, Z. Beata, Van den Nest Miriam, E. Julia, P. Elisabeth, and D.-E. Magda, "The use of a UV-C disinfection robot in the routine cleaning process: a field study in an Academic hospital", *Antimicrob Resist Infect Control*, vol. 10, no. 1, pp. 1–10, 2021. doi: 10.1186/s13756-021-00945-4

[12] C. McGinn, R. Scott, N. Donelly, K. L. Roberrts, M. Bogue, C. Kiernan, M. Beckett, "Exploring the applicability of robot-assisted uv disinfection in radiology", *Front Robot AI*, vol. 7, pp. 1–12, 2021. doi: 10.3389/frobt.2020.590306

[13] A. Simarmata, T. K. Dewi, M. R. Fikri, and K. Saptaji, "Development of autonomous robot using fuzzy inference system (FIS) for indoor ultraviolet (UV) disinfection," in *Proceedings of The Symposium on Advance of Sustainable Engineering 2021: Post Covid-19 Pandemic: Challenges and Opportunities in Environment, Science, and Engineering Research*, pp. 1–7, 2023. doi: 10.1063/5.0112788

[14] A. Dairi, F. Harrou, Y. Sun, and M. Senouci, "Obstacle detection for intelligent transportation systems using deep stacked autoencoder and k -nearest neighbor scheme," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5122–5132, 2018. doi: 10.1109/JSEN.2018.2831082

[15] X. Gao, H. Wu, L. Zhai, H. Sun, Q. Jia, Y. Wang, and L. Wu, "A rapidly exploring random tree optimization algorithm for space robotic manipulators guided by obstacle avoidance independent potential field," *International Journal of Advanced Robotic Systems*, vol. 15, no. 3, pp. 1–17, 2018. doi: 10.1177/1729881418782240

[16] I. S. Asti, T. Agustinah, and A. Santoso, "Obstacle avoidance with energy efficiency and distance deviation using knn algorithm for quadcopter," in *2020 International Seminar on Intelligent Technology and Its Applications*, pp. 285–291, 2020. doi: 10.1109/ISITIA49792.2020.9163788

[17] B. Ilias, S. A. A. Shukor, A. H. Adom, N. A. Rahim, M. F. Ibrahim, and S. Yaacob, "Indoor mobile robot localization using KNN," in *2016 6th IEEE International Conference on Control System, Computing and Engineering*, pp. 211–216, 2016. doi: 10.1109/ICCSCE.2016.7893573

[18] N. J. Zakaria, H. Zamzuri, M. H. Mohamed Arif, M. Z. Azmi, and N. Hassan, "Lane change behaviour recognition using neural network," *Journal of Mechanical Engineering*, vol. 6, no. 1, pp. 1–19, 2018.

[19] E. Sochima Vincent, E. Chidi, and Engr. Dr. P. Ene, "Deep artificial neural network based obstacle detection and avoidance for a non-holonomic mobile robot," *International Research Journal of Applied Sciences, Engineering, and Technology*, vol. 5, no. 1, pp. 1–15, 2019.

[20] V. Andreev and V. Tarasova, "The mobile robot control for obstacle avoidance with an artificial neural network application," *30th DAAAM International Symposium On Intelligent Manufacturing And Automation*, pp. 0724–0732, 2019. doi: 10.2507/30th.daaam.proceedings.099

[21] G. Farias, E. Fabregas, E. Oeralta, H. Vargas, G. Hermosilla, G. Garcia, and S. Dormido, "A neural network approach for building an obstacle detection model by fusion of proximity sensors data", *Sensors*, vol. 18, no. 3, pp. 1–18, 2018. doi: 10.3390/s18030683

[22] A. Medina-Santiago, L. A. Morales-Rosales, C. A. Hernández-Gracidas, I. Algredo-Badillo, A. D. Pano-Azucena, and J. A. Orozco Torres, "Reactive obstacle–avoidance systems for wheeled mobile robots based on artificial intelligence," *Applied Sciences*, vol. 11, no. 14, pp. 1–21, 2021. doi: 10.3390/app11146468

[23] K. K. A. Farag, H. H. Shehata, and H. M. El-Batsh, "Mobile robot obstacle avoidance based on neural network with a standardization technique," *Journal of Robotics*, vol. 2021, pp. 1–14, 2021. doi: 10.1155/2021/1129872

[24] V. Miranda, A. Neto, G. Freitas, and L. Mozelli, "On the generalization of deep reinforcement learning methods in the problem of local navigation", *ArXiv*, vol. 14271, pp. 1–17, 2022.

[25] Ş. Yıldırım and S. Savaş, "Design of a mobile robot to work in hospitals and trajectory planning using proposed neural networks predictors," in

International Conference on Reliable Systems Engineering, vol. 305, pp. 32–45, 2022. doi: 10.1007/978-3-030-83368-8_4

[26] A. J. Abougarair, M. K. I. Aburakhis, and M. M. Edardar, "Adaptive neural networks based robust output feedback controllers for nonlinear systems," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 37–56, 2022. doi: 10.31763/ijrcs.v2i1.523

[27] C. El Mechal, N. E. A. El Idrissi, and M. Mesbah, "CNN-based obstacle avoidance using RGB-depth image fusion," in *WITS 2020*, vol. 745, pp. 867–876, 2022. doi: 10.1007/978-981-33-6893-4_78

[28] S. Duan, H. Ma, J. Zhang, and G. Liu, "Development of an automatic lawnmower with real-time computer vision for obstacle avoidance," *International Journal of Computational Methods*, vol. 19, no. 8. 2021, doi: 10.1142/S0219876221420019.

[29] L. Ma, W. Xie, and H. Huang, "Convolutional neural network based obstacle detection for unmanned surface vehicle", *Mathematical Biosciences & Engineering*, vol. 17, no. 1, pp. 845–861, 2019. doi: 10.3934/mbe.2020045

[30] Z. Zhang, M. Xiong, and H. Xiong, "Monocular depth estimation for UAV obstacle avoidance," in *2019 4th International Conference on Cloud Computing and Internet of Things (CCIOT)*, pp. 43–47, 2019. doi: 10.1109/CCIOT48581.2019.8980350

[31] N.-H. Chang, Y.-H. Chien, H.-H. Chiang, W.-Y. Wang, and C.-C. Hsu, "A robot obstacle avoidance method using merged CNN framework," in *2019 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 1–5, 2019. doi: 10.1109/ICMLC48188.2019.8949168

[32] S. Che Abdullah, M. Ohka, and M. D. Amari, "Stereo vision for visual object tracking and distance measurement assessment," *Journal of Mechanical Engineering*, vol. 16, no. 1, pp. 121–134, 2019. doi: 10.24191/jmeche.v16i1.6043

[33] M. K. Osman, M. E. A. Mohammad Zamree, M. Idris, and K. A. Ahmad, "Pavement crack classification using deep convolutional neural network," *Journal of Mechanical Engineering*, vol. 10, no. 1, pp. 227–244, 2021

[34] A. Sarda, S. Dixit, and A. Bhan, "Object Detection for Autonomous Driving using YOLO algorithm," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, pp. 447–451, 2021. doi: 10.1109/ICIEM51511.2021.9445365

[35] A. Karne, R. Karne, V. K. Kumar, and A. Arunkumar, "Convolutional neural networks for object detection and recognition," *Journal of Artificial Intelligence, Machine Learning and Neural Network*, vol. 3, no. 3, pp. 1–13, 2023. doi: 10.55529/jaimlnn.32.1.13

[36] X. Li, Z. Xie, X. Deng, Y. Wu, and Y. Pi, "Traffic sign detection based on improved faster R-CNN for autonomous driving," *The Journal of*

Supercomputing, vol. 78, no. 6, pp. 7982–8002, 2022. doi: 10.1007/s11227-021-04230-4

[37] J. Yu, Z. Chen, C. Wu, and X. Cheng, "Intelligent vehicle road type recognition based on mask R-CNN," in *20th COTA International Conference of Trnsportation Professional*, pp. 942–951, 2020. doi: 10.1061/9780784483053.079

[38] J. Yang, Y. Wang, H. Zhao, and G. Gui, "MobileNet and knowledge distillation-based automatic scenario recognition method in vehicle-to-vehicle systems," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 10, pp. 11006–11016, 2022. doi: 10.1109/TVT.2022.3184994

[39] H. Lee, I. Ullah, W. Wan, Y. Gao, and Z. Fang, "Real-Time vehicle make and model recognition with the residual squeezenet architecture," *Sensors*, vol. 19, no. 5, pp. 1–14, 2019. doi: 10.3390/s19050982

[40] M. Oktaviandri and D. K. A. V Paramasivam, "Design and fabrication of customized ais kacang vending machine," *Indonesian Journal of Computing, Engineering and Design (IJoCED)*, vol. 2, no. 1, pp. 1–8, 2020. doi: 10.35806/ijoced.v2i1.100

[41] S. Rahardian, I. Putra, and B. Budiman, "On the use of steel and aluminum materials for frame structure of electric trike," *Indonesian Journal of Computing, Engineering and Design (IJoCED)*, vol. 3, no. 1, pp. 9–18, 2021. doi: 10.35806/ijoced.v3i1.122

[42] S. M. LaValle, "Motion planning: The essentials", *Robotics & Automation Magazine, IEEE*, vol. 18, no. 1, pp. 79–89, 2011. doi: 10.1109/MRA.2011.940276

[43] M. M. Taye, "Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions", *Computation*, vol. 11, no. 3, p. 52, 2023. doi: 10.3390/computation11030052