UDC 614.841.34:681.5:004.89
DOI: 10.15587/2706-5448.2022.269861

Amin Saif,
Gamal Muneer,
Yusuf Abdulrahman,
Hareth Abdulbaqi,
Aimen Abdullah,
Abdullah Ali,
Abduljalil Derhim

DEVELOPING A PROTOTYPE OF FIRE DETECTION AND AUTOMATIC EXTINGUISHER MOBILE ROBOT BASED ON CONVOLUTIONAL NEURAL NETWORK

The object of research is a prototype of fire detection and automatic extinguisher mobile robot based on convolutional neural network. Within the recent few decades, fires are considered as one of the most serious disaster that occurs in many places around the world. The severity of fire incidents causes damages to buildings, infrastructures and properties. Resulting losses of human's life and costs them a lot of losses. Thus, fire poses a great threat to us significantly; it is extremely dangerous for fire fighters. Fires can be resulted by materials such as rubber and chemical products. Other sources of fire are the short circuits on electrical devices and faults in power circuits. Additionally, overheating and overloading problems can be the cause of fire incidents. All these reasons lead to bad consequences when there is no immediate response to such problems. The advent of computer vision technology has played such a significant role for human life. Artificial intelligence field has improved the efficiency and behaviors of robotics beyond expectations. The interference of artificial intelligence made robotics act intelligently. For this reason, in this paper we presented a mobile robot based on deep learning to detect the fire source and determines its coordinate position then automatically moves toward the target and extinguish fire. Deep learning algorithms are the efficient ones for object detection applications. CNN model is one of the most common deep learning algorithms which have been used in the study for the fire detection. Due to the insufficient amount of datasets and large efforts required to build model from scratch. MobileNet V2 is one of the CNN models that support transfer learning technique. After training the model and testing it on 20 % of the used datasets the classification accuracy achieved up to 98.01 %. The motion repeatability of the robot has been implemented and tested resulting mean error 0.648 cm.

Keywords: *mobile robot, supervised deep learning, convolutional neural network (CNN), image processing, transfer learning.*

Received date: 05.12.2022
Accepted date: 21.12.2022
Published date: 27.12.2022

How to cite
Saif, A., Muneer, G., Abdulrahman, Y., Abdulbaqi, H., Abdullah, A., Ali, A., Derhim, A. (2022). Developing a prototype of fire detection and automatic extinguisher mobile robot based on convolutional neural network. Technology Audit and Production Reserves, 6 (1 (68)), 15–23. doi: https://doi.org/10.15587/2706-5448.2022.269861

1. Introduction

Fire disaster is one of the most destructive issues in our life [1]. Fire has been gaining momentum and reaching alarming proportions [2]. Resulting losses of people life, surrounding environments, buildings, infrastructure, commercial and businesses [3]. The wide spread and severity of fire incidents has caused a domino effect in the buildings which becomes such a prominent issue in the recent years [3]. Fire disaster poses a great threat to people as well as fire fighters when there is no immediate responding for initial stages of fire [4]. Firefighting is also an extremely dangerous mission [5–7]. And is difficult [6, 8]. Specially, when it comes to dealing with unknown environments. The goal of the paper is to develop a prototype of automatic mobile robot to detect fire using deep learning for image classification of fire and non-fire classes. Then automatically moves toward fire to extinguish it. The developed system is controlled via main controller PC for image processing and secondary controller installed on mobile robot that has motion control algorithm. The system includes webcam of 1 megapixel to capture the fire image and the PC shows the frames and reference frame ground, temperature sensor is used to check the temperature increment, and Bluetooth module as a means for wireless communication. L298N driver module is used to provide the dc motors with the required current for the mobile robot motion. The robot works in a 3×3 squared meter area. This area must be clean with no obstacles for better operation.

The different methods and ideas previous researchers followed to developed fire extinguishing mobile robot. Some of these papers are.

The developed system called AFFMP in the paper [9]. The system has flame sensor for fire detection. Additionally, it follows a guiding path to avoid obstacles that hinder robot from movement. The commands of detection are sent to the microcontroller to start extinguishing fire using water pump.

The proposed of autonomous firefighting robot system in the paper [10]. The system has been developed to aid firefighting robot that encounters difficulties in detecting fire with an environment filled with smoke. The aim objective is to enhance system stability in real time processing. The proposed system is provided with two stereo thermal infrared (IR) visions for fire detection and a frequency modulated continuous wave (FMCW) for position determination.

The voice controlled intelligent fire extinguisher robot in the paper [11]. The system is controlled by voice commands given by human that includes direction movement. The system has flame sensor which will detect fire flame, camera to show video fire on LCD screen, buzzer for alarm warning, CO_2 blower to extinguish it, PIC16F77A microcontroller and IR sensor is used to avoid obstacle by the user when barriers is found in front of robot.

In the firefighting robot in the paper [12] a web server is used for handling the robot from web page and can be able to monitor different parameters from web server. The system used conventional detectors for fire detection. The motion is controlled via android phone.

The autonomous firefighting robot was developed with self-power management in the paper [13].

The mobile robot with sensor fusion fire detection unit was developed in the paper [14].

The proposed system «Q ROB» in the paper entitled development of firefighting robot [15]. It is designed to rescue firefighters from dangerous situations that may encounter in narrow places. The robot also has ultrasonic transducer to avoid obstacles. Operator monitors the robot motion via the installed camera and controls the robot to extinguish fire.

The proposed system DTMF centered remotely located fire extinguishing robot in the paper [16]. The robot is controlled via mobile phone for controlling movement of robot responding to flame sensor. The user sends commands known as DTMF tone to the robot. The designed embedded system with a decoder converts transmitted signal to binary bits that each command is specialized for particular task.

The firefighting robot based on deep learning for fire detection developed in the paper [17]. The system combined Alexnet model and Imagenet for fire detection after training and testing the model. The accuracy achieved up to 98.25 % of classification. The system detects fire by PI camera then sends information to Raspberry pi microprocessor for controlling the mobile robot motion.

Deep learning (DL) is the most common subfield of machine learning algorithms which has the ability to perform well compared with conventional machine learning algorithms [18, 19]. It is used in this study because it deals with a huge amount of data [18, 20]. Additionally, it has good performance and simplicity in processing and analyzing datasets significantly in image classification tasks [21, 22]. The output results are mapped with the input data resulting a function from input to output [23, 24]. It is considered as the backbone and the basic field for dealing with variety of tasks of machine learning [25, 26]. Convolutional neural network (CNN) is described as one of the deep learning neural networks that are used for processing the content of images and processing complex tasks using convolution operation [27–29]. It is the most popular network algorithm compared with the other DL neural networks [30, 31]. What distinguish CNN are the filters it uses to generate features from a large amount of datasets automatically and learn these features to classify datasets [32, 33]. In deep learning algorithms, it's necessary to have a huge amount of training datasets to get a good performance model [34]. Thus, building trials from scratch becomes traditional way. Additionally, it takes long time to collect data from zero and finally ending with insufficient amount of datasets [35]. Therefore, with the aid of transfer learning technique that participates in CNN models with a huge amount of data. This method allows for transferring knowledge into the new generated desired model using optimized updated weights and bias to the learner during training process [36]. Transfer learning technique boosted the efficiency and performance to deal with recognition and computer vision based fields [37, 38].

2. Materials and Methods

The object of research is a prototype of fire detection and automatic extinguisher mobile robot based on convolutional neural network.

MobileNet V2 is one of the convolution neural network models that were used in the study due to its good performance and its simplicity. The basic architecture of the model is the inverted residual in which the input and the output are narrow layers named as bottleneck layers where residual connections between these layers as in Fig. 1.

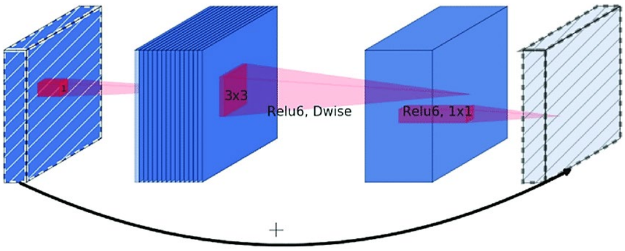

Fig. 1. Inverted residual block

The model has 32 convolutional filters and 19 residual bottleneck layers. However, the architecture is different from conventional models, which work on expanding dimensionality in the input. The model applies a bottleneck convolution to the input to change the dimensionality from small dimensions that contains small channels to a larger dimension with a high number of channels, which is done by using linear convolution that retrieves features to a low dimension. The architecture of model strives to minimize latency of smaller scale network so that computer vision applications run well on mobile devices. This is achieved by minimizing the required memory and also the number parameters while maintaining the same accuracy. The architecture of the model with different strides is shown in Fig. 2. The architecture with stride 1 refers that residual connection is not required while the architecture with stride 2 requires residual connection.

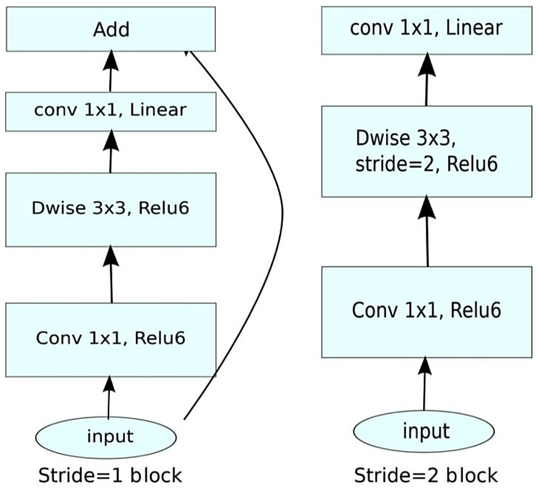

Fig. 2. Architecture of CNN model

What distinguishes the module and enhances the efficiency of the model is that the model architecture includes depthwise separable convolution between bottleneck layers rather than a full convolution operator which splits the convolution into two convolutions to reduce the number of times of multiplications. So that takes long time for computation relative to addition operation. Thus, depth wise convolution reduces the computation time and parameters. According to [39] Table 1 compares the performance between MobileNets and and MobileNet V2.

Table 1

Comparsion of MobileNet and MobileNet V2 models

Model	ImageNet Accuracy	Million Mult-Adds	Million Parameters
Conv MobileNet	71.7 %	4866	29.3
MobileNet	70.6 %	569	4.2

It turns out the accuracy on image net drops only 1 % while using significantly less number of parameters from 29.3 million the number of parameters it is down to just 4.2 million. Thus, MobileNet V2 model was used in this study.

The proposed method for the study is divided into three main stages: software design, mechanical design and electrical design.

2.1. Software design. Computer vision system design show in Fig. 3.

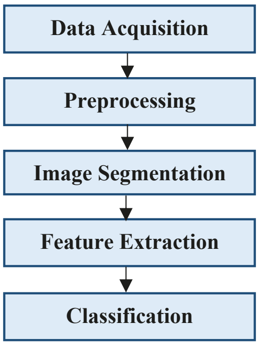

Fig. 3. Image Processing Steps

2.1.1. Datasets. The datasets used in the study were 755 images. They were classified as 604 for training and 151 for validation. They were captured using smartphones as (2268*4032) JPEG files of size (224,224,3). The images were captured from different angles, directions and different lightning conditions. They were also captured from different distances and various sizes in both indoor and outdoor environments. And most importantly, they were in good quality. The images consist of two classes named fire and non-fire images. The process took 3 days for capturing images. The reason behind using captured images rather than using data science images is that the study will be applied in real world. Thus, datasets required to be as real world as much as possible to predict better results for real time process.

2.1.2. Preprocessing. In this process, the fire detection process will be exposed to noises and influences in external and internal environment. The source of noises such as lighting effect, illumination and many gradations of color between fire color and non-fire image color. The aim of this process is to enhance and improve images by reducing the noises on images by applying filtering and minimizes the False positive predictions.

2.1.3. Feature Extraction. The determination of certain pixels within a fire image pixel. The represented pixels are desired and will be extracted from the image.

2.1.4. Image Segmentation. The process that divides the image into multiple objects of the same interest. It isolates the background from an object in the object image. This causes difficulty to the process. The aim of segmentation process is to minimize the complexity of an image so that it becomes more understandable and simple for analysis. Essentially, it assigns labels into the pixels of an image where the other pixels in the image have common properties then they belong to the same labeled group. Image segmentation is a part of image classification. The image classification identifies the class, which the image belongs to, while image segmentation identifies the class which the elements of an image belongs.

2.1.5. Training and Hyper-Parameters. The Hyperparameters used when training the model on the same dataset, shown in Table 2.

Table 2

Hyperparameters during training

Hyper-parameter	CNN Model
Learning Rate	0.0001
Epoch	20
Batch Size	32
Optimizer	Adam
Loss Function	Binary Categorical Cross Entropy
Dropout	0.5

2.1.6. Software Libraries. *Tensorflow* is used in the field of machine learning. It includes a thorough, scalable and flexible ecosystem of tools, libraries and community resources to build machine-learning applications.

Keras is used by researchers for developing and evaluating deep learning neural network models.

Matplotlib is python library used for 2D graphical representation and data science visualization.

Pandas is a python open source library used for analyzing and processing data.

Numpy is a python manipulate multi-dimensional arrays observation for numerical python open source library that basically works with arrays.

Scikit-Learn is a python open source machine learning library that has been developed to support libraries like Numpy, Scipy and Matplotlib. The library includes efficient tools and statistical modeling like clustering, classification and regression.

Seahorn is a well visualization python library used to produce good styles for plotting and graphics in an attractive plot.

Random is a module that is an extension for python features used to generate random numbers.

Argparse is a standard library in python used to automatically integrate the parsing of commend line arguments.

2.2. Mechanical design. The mobile robot structure has been designed using computer aided design and all parts were assembled and Fig. 4 shows are the overall model of mobile robot.

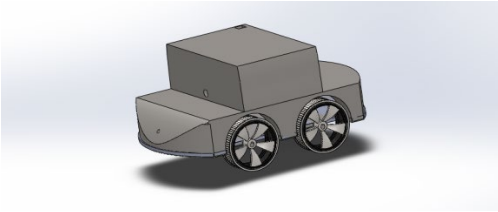

Fig. 4. Assembly Model in SolidWorks

2.2.1. Kinematic Modelling. The Kinematic equations of the mobile robot will be obtained to determine the transformation matrix of the target frame of robot relative to the reference frame of the camera. This is done when the following assumptions are met:

Assumption 1. As shown in Fig. 5 the mobile robot is symmetrical along lateral and longitudinal axes.

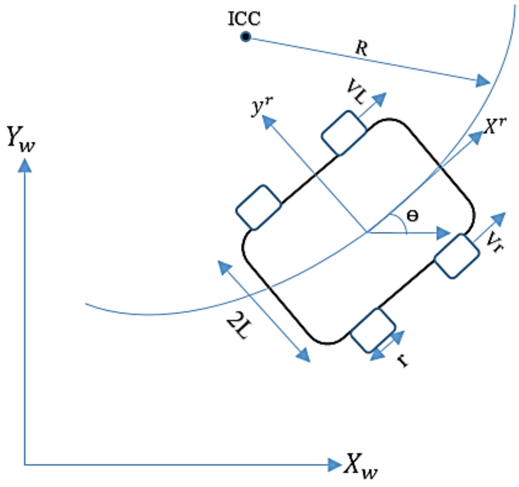

Fig. 5. Schematic of Mobile Robot in Coordinate Systems

The load above carrier board such as electronic boards, water container is assumed to be concentrated on the mid-point of the carrier board. Therefore, the center mass of the robot is assumed to be located at the origin point in the carrier board, the mid-point is denoted as C.

Assumption 2. The mobile robot does not move sideways which implies that the mobile robot does not have a lateral component $\dot{X}^r = 0$ this will provide a curved motion called instantaneous center of curvature (ICC).

Assumption 3. The robot moves without slippage, which indicates that for every full revolution $\{\Delta\varnothing = 2\pi\}$ each wheel travels a distance equal to its circumference $\{\Delta X = 2\pi R\}$. This also means that the four wheels are always at a contact point on the ground. At the Instantaneous Center of Curvature (ICC) all points in a pure rotation field centered at ICC have velocity orthogonal to the distance of radius of ICC. Thus,

$$V_L = \omega(R - L), \tag{1}$$

$$V_C = \omega R, \tag{2}$$

$$V_R = \omega(R + L), \tag{3}$$

where R represents the radius of Instantaneous Center of Curvature; ω is the angular velocity of the mobile robot body; V_C is the linear velocity at the center point; V_L is the net linear velocity of the left wheels; V_R is the net linear velocity of the right wheels. To clarify the motion of the mobile robot there are some exceptions of motion can occur only when the model satisfies the conditions of kinematic constraints.

This kinematic describe how the mobile robot will move resulting in a specific position and orientation when given a sequence of commands to the wheels $\varnothing_1, \varnothing_2 \cdots \varnothing_i$ additionally, it also describes the linear and angular velocities the robot moves in.

From the equations (4), (5) and (6) of the linear velocities of the wheels. By Adding both equations (4) and (6), then the result is the velocity at the center point which has new formula:

$$V_{Cx} = \frac{V_R + V_L}{2} = \frac{R}{2}\left(\dot{\varnothing}_r + \dot{\varnothing}_L\right). \tag{4}$$

By subtracting equation (6) from (4) the result is the angular velocity of the robot which has the following formula:

$$\omega = \frac{V_R - V_L}{2L} = \frac{R}{2L}\left(\dot{\varnothing}_r - \dot{\varnothing}_L\right). \tag{5}$$

The kinematic equation of velocities in the local frame is given by the following vector of three components:

$$\dot{P}_R = \begin{bmatrix} \dot{X}_C \\ \dot{Y}_C \\ \dot{\theta}_C \end{bmatrix} = \begin{bmatrix} V_{Cx} \\ V_{Cy} \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2}\left(\dot{\varnothing}_r + \dot{\varnothing}_L\right) \\ 0 \\ \frac{R}{2L}\left(\dot{\varnothing}_r - \dot{\varnothing}_L\right) \end{bmatrix}. \tag{6}$$

Based on kinematic constraints conditions there is no lateral velocity component $V_{Cy} = 0$. By using the velocity propagation method to obtain the Jacobian matrix by taking the partial differentiation for the three parameters of V with respect to $\dot{\varnothing}_r$ and $\dot{\varnothing}_L$ for equation (6):

$$\dot{P}_R=\begin{bmatrix}\frac{\partial v_{cx}}{\partial\dot{\varnothing}_r} & \frac{\partial v_{cx}}{\partial\dot{\varnothing}_L}\\ \frac{\partial v_{cy}}{\partial\dot{\varnothing}_r} & \frac{\partial v_{cy}}{\partial\dot{\varnothing}_L}\\ \frac{\partial\omega}{\partial\dot{\varnothing}_r} & \frac{\partial\omega}{\partial\dot{\varnothing}_L}\end{bmatrix}\begin{bmatrix}\dot{\varnothing}_r\\ \dot{\varnothing}_L\end{bmatrix}. \quad (7)$$

$$\dot{P}_R=\begin{bmatrix}\frac{r}{2} & \frac{r}{2}\\ 0 & 0\\ \frac{r}{2L} & \frac{-r}{2L}\end{bmatrix}\begin{bmatrix}\dot{\varnothing}_r\\ \dot{\varnothing}_L\end{bmatrix}. \quad (8)$$

The Jacobian matrix of velocities is expressed in the following matrix:

$$J=\begin{bmatrix}\frac{r}{2} & \frac{r}{2}\\ 0 & 0\\ \frac{r}{2L} & -\frac{r}{2L}\end{bmatrix}. \quad (9)$$

From the property in formula (2). Based on the property, which states that for orthogonal rotation matrices, the inverse of rotation matrix is equal to rotation matrix.

$$\dot{P}_W=R(\theta)^{-1}\dot{P}_R. \quad (10)$$

$$R(\theta)=R(\theta)^{-1}=\begin{bmatrix}C\theta & -S\theta & 0\\ S\theta & C\theta & 0\\ 0 & 0 & 1\end{bmatrix}. \quad (11)$$

$$\dot{P}_w=\begin{bmatrix}C\theta & -S\theta & 0\\ S\theta & C\theta & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\frac{r}{2} & \frac{r}{2}\\ 0 & 0\\ \frac{r}{2l} & \frac{-r}{2l}\end{bmatrix}\begin{bmatrix}\dot{\varnothing}_r\\ \dot{\varnothing}_l\end{bmatrix}. \quad (12)$$

$$\dot{P}_W=\begin{bmatrix}\dot{X}_\omega\\ \dot{Y}_\omega\\ \dot{\theta}_\omega\end{bmatrix}=\frac{r}{2}\begin{bmatrix}C\theta & C\theta\\ S\theta & S\theta\\ \frac{1}{L} & \frac{-1}{L}\end{bmatrix}\begin{bmatrix}\dot{\varnothing}_r\\ \dot{\varnothing}_l\end{bmatrix}. \quad (13)$$

Equations (14), (15) and (16) represent the kinematic equations in terms of angular and linear velocities in the body frame with respect to the global frame. The position and orientation of the mobile robot current pose in the global frame is obtained by taking the integration from 0 to t. The mobile robot is desired to rotate at 90 degrees using skid steering method, which depends on difference in velocities. For this orientation, the difference in the velocities must be 16 rad/s. This is done by adjusting the duty cycle for the DC motors that is obtained by the pulse width for the time in equation (16):

$$X(t)=\int_0^t\frac{(\dot{\varnothing}_r+\dot{\varnothing}_l)r\cos(\theta)dt}{2}. \quad (14)$$

$$Y(t)=\int_0^t\frac{(\dot{\varnothing}_r+\dot{\varnothing}_l)r\sin(\theta)dt}{2}. \quad (15)$$

$$\theta(t)=\int_0^t\frac{(\dot{\varnothing}_r-\dot{\varnothing}_l)rdt}{2l}. \quad (16)$$

2.2.2. Mobile robot simulation using simscape multibody. This tool is used to simulate models of 3D mechanical systems like robotics, manipulators and many others. It allows to provide modelling with the use of blocks that identifies the mechanical parts such as links, joints, bodies. The aim of this tool is to find the building blocks for the motion of the mechanical systems. The procedures can be performed starting with the design of the mechanical systems using CAD. Then selecting appropriate properties for the model such as material, masses, constraints. Then by using Simscape multibody link to export the file into xml. The xml file is imported from Matlab using the function smimport. The simulation of the mechanical system will be shown. The Simscape was used in the study to ensure that all components of robot such as wheels and body frame will move with respect to the reference frame. The coordinate frames have been attached to each part of the mobile robot before importing the Simscape model. Fig. 6, which shows the building block for the mobile robot motion and Fig. 7, shows the mobile robot in Matlab Simulink.

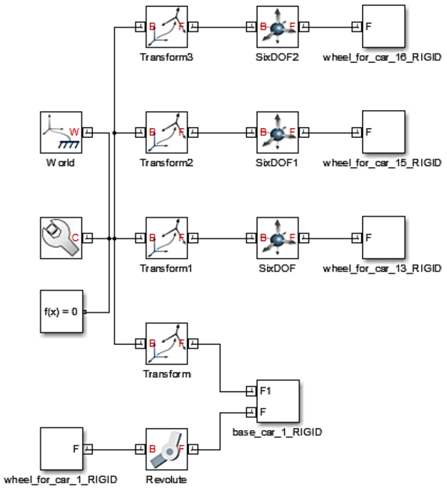

Fig. 6. Building block for mobile robot motion

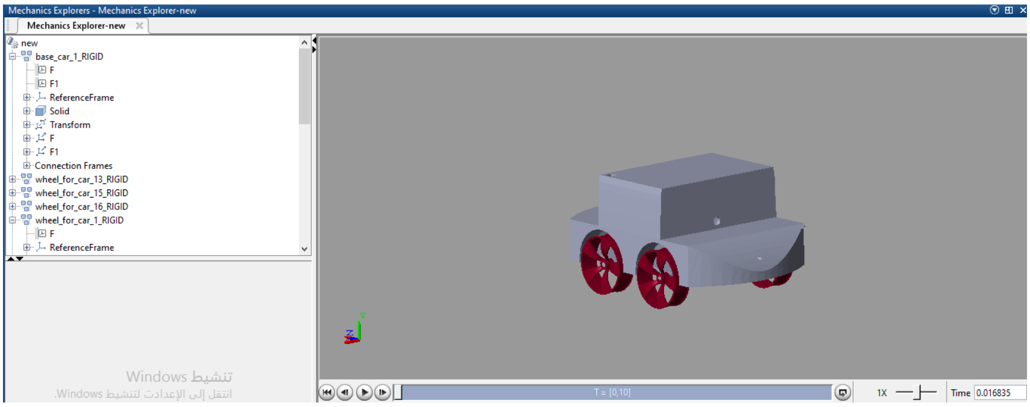

Fig. 7. Mobile robot in MATLAB Simulink

2.3. Electronic circuit design. Electronic components considered as an important part in designing Mechatronics projects, for our project we have chosen the electronic parts precisely according to their characteristics and quality of work. It consists of a webcam, an Arduino Uno, DC motors, drivers, Bluetooth module, water pump, power supply and LM35 sensor. All electronic components were connected and simulated and Fig. 8 represents the overall electronic circuit wiring.

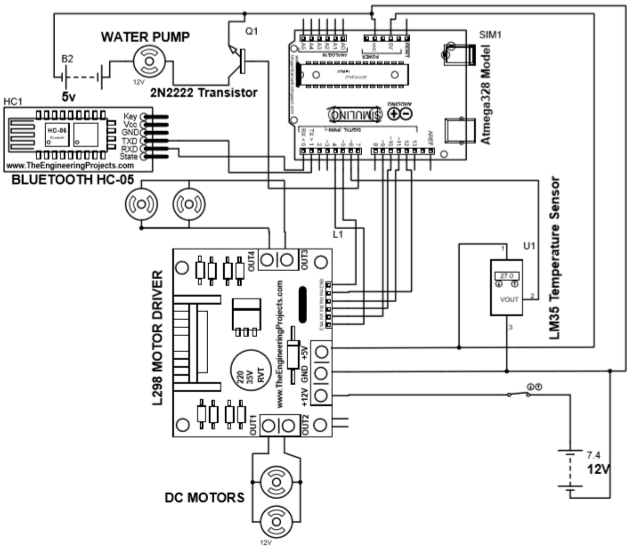

Fig. 8. Electronic circuit connection

3. Results and Discussion

3.1. Evaluation metric. The evaluation related to deep learning and plays such a significant part for achieving a better classification model. It is employed with data classification process through two steps training and testing. It used to optimize the classification in the training step to evaluate the efficiency of resulted classification model. There is evaluation metrics discussed below.

3.2. Confusion matrix. The confusion matrix is defined as a table that is used to measure how well the classification model. It can perform on a set of test data for which true values are known. There are four main terms which confusion matrix is based on. The terms may be confusing but makes confusion matrix more comprehended true positive (*TP*), true negative (*TN*), false positive (*FP*) and false negative (*FN*) as shown in Fig. 9.

Accuracy is obtained by the ratio of summation values of correct classification samples to the total number of classes:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}. \qquad (17)$$

Recall is also called sensitivity or true positive rate (TPR) which indicates how the classifier correctly predicted the classes:

$$Recall = \frac{TP}{TP+FN}. \qquad (18)$$

Precision is an evaluation metric that determines the number of correct positive predicted samples. It obtains the accuracy of minority class. It's obtained by the ratio of number of correct predictions to the all number of positive predictions:

$$Precision = \frac{TP}{TP+FP}. \qquad (19)$$

*F*1-*Score* is an evaluation metric based on the combination of precision and recall evaluations of the model. It takes the harmonic mean of the precision and recall:

$$F1\text{-}Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}. \qquad (20)$$

According to the confusion matrix in Fig. 9, the results of the evaluation metrics were obtained and given in Table 3.

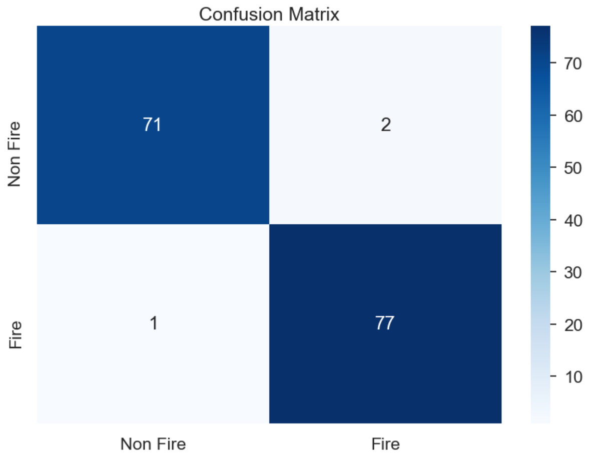

Fig. 9. Confusion matrix for model

Table 3

Evaluation Metrics of CNN Module

Evaluation Metrics	Model
Accuracy	98.01 %
Precision	97.47 %
Recall	98.72 %
F1-Score	98.09 %

Fig. 10 shows the validation and training accuracy curves of the model after training the model with epochs of 20.

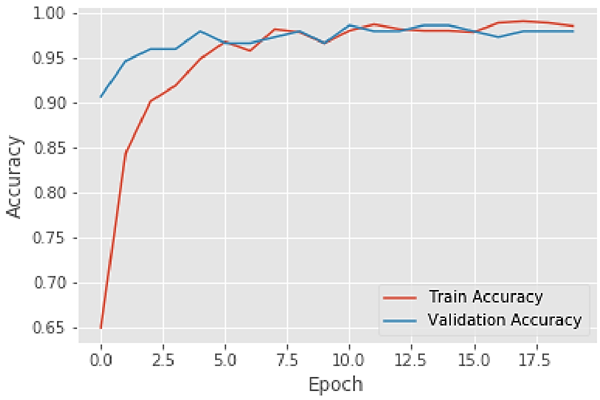

Fig. 10. Training and validation accuracy of CNN model

The curve shows that both train and test data are very close together. However, the curves are not perfectly smooth, they turned out to have regular shape. Although, there are methods were used to overcome overfitting and to optimize the accuracy curve better such as dropout function, data augmentation techniques and Adam optimizer. However, the epochs value have been selected once and resulting in this curve.

Fig. 11 shows the relation between the reductions of Loss function with increment of epoch. The epoch was set to 20. Both train and validation loss curves are very close together. In this case, the learning rate is very low resulting to the reduction of the cost function.

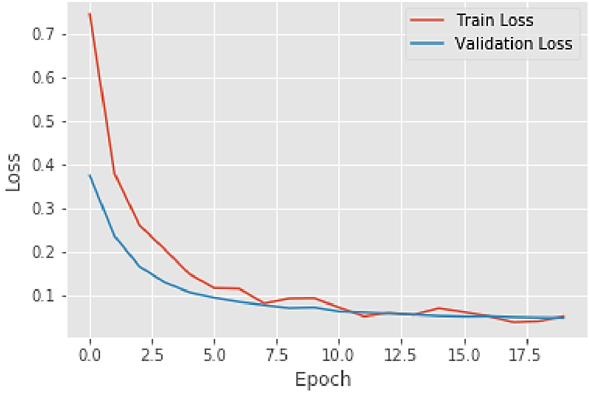

Fig. 11. Training and validation loss of CNN model

The carrier board of the mobile robot has been designed using and manufactured using Laser Cutting Machine. The board is a MDF type of composite materials, which has been machined, in a desired dimension. Fig. 12 shows the vector file for the mobile robot design.

The electronic circuit including Atmega 328 module, temperature sensor IC, switch and Bluetooth module that were connected correctly using wires to transmit the electrical power and cables to transmit data to and from the microcontroller and then installed to the mobile robot as shown in Fig. 13, 14.

Fig. 15 shows the experimentation of fire detection. This experiment shows that fire detected in different times and different environments at the existing of light and the absence of light. The 2-dimensional coordinate positions.

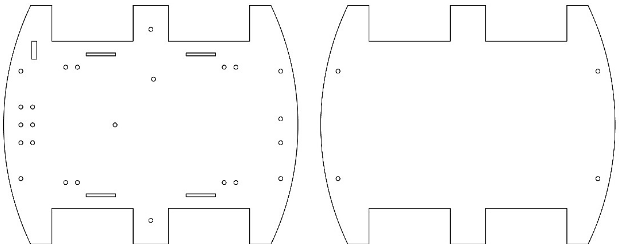

Fig. 12. Carrier design vector file

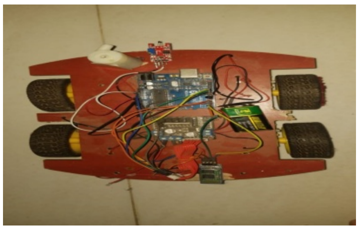

Fig. 13. Electronic circuit connection

Fig. 14. Mobile robot

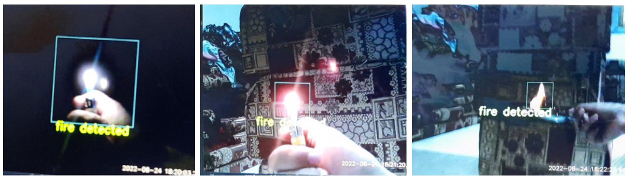

Fig. 15. Fire detection test

Table 4 shows the results of the mobile robot repeatability test.

Table 4

Inverse kinematics test results

Xin (cm)	Yin (cm)	Resultant distance	Actual distance	Error
10	10	14	14.14	0.14
20	20	27.6	28.28	0.68
30	30	41.5	42.43	0.93
40	40	56	56.57	0.57
50	50	70	70.71	0.71
60	60	83.9	84.85	0.95
70	70	98.2	98.99	0.79
80	80	112.7	113.14	0.44
90	90	126.93	127.28	0.35
100	100	140.5	141.42	0.92
Average				0.648

The test was implemented in the following manner: ten coordinate positions were selected in alignment on the floor. The coordinates of each point given as input to the inverse kinematic algorithm. An error was calculated by calculating the absolute distance between the measured resultant value the robot has moved to and the expected value. The measured value was measured from the midpoint of the mobile robot and the reference position. Then, after the test was carried out for all points. The mean error for the robot for the ten test points was 0.648 cm.

The proposed system was implemented for fire detection using pretrained CNN model. As shown in Fig. 8 which represents the confusion matrix for fire detection model. The transfer learning from the pretrained MobileNet V2 model was used to train the model. The total images were 151 images, for the two classes 79 for fire class and 72 for non-fire class. The confusion matrix also shows that two images of fire class were predicted as non-fire, which indicates two false negative predictions. One image of non-fire class was predicted as fire, which indicates one false positive prediction. The number of true positive states were 77 while the number of true negative states were 71. Therefore, they were tested using binary classifier. The accuracy of detection was 98.01 %. The study was designed as a prototype, so a few of datasets were used to avoid wasting time and efforts. The robot was producing shifting distance from real position. Therefore, correction factor was used in the program to adjust the motion precision of the mobile robot and optimize its inverse kinematic. After adjusting precision, the selected test points shown in Table 4 was used to determine the mean error of robot. The result shows that robot perform better repeatability with a mean error of 0.648 cm.

4. Conclusions

This study presents a prototype of an automatic fire detection mobile robot using pretrained deep learning model as well as an extinguisher unit for fighting fire source. The result shows systems based on computer vision operates more efficiently than systems with conventional sensors. The pretrained CNN model used for training model performs better performance for fire detection systems. The training process takes less time and they are fast in detecting in real time. The CNN model was trained and the performance was evaluated. The frames were considered to convert transformation matrix of the body frame of robot relative to the reference frame of the camera. The system was implemented and achieved all the objectives successfully. It takes some time to make this system much better for performing complex tasks. An ultrasonic transducer should use for the robot to avoid terrain rough surfaces. Smoke sensor should be used to improve the robot's efficiency. Increasing the number of datasets is recommended to improve the accuracy of detection. The base of robot is recommended to be made of metal to withstand heat when extinguishing fire.

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research and its results presented in this paper.

Financing

Presentation of research in the form of publication through financial support in the form of a grant from SUES (Support to Ukrainian Editorial Staff).

Data availability

The manuscript has no associated data.

References

1. Brushlinsky, N. N., Ahrens, M., Sokolov, S. V., Wagner, P. (2016). World fire statistics. *Cent. fire statistics.,* 10.
2. Woodrow, B. (2012). *Fire as Vulnerability: The Value Added from Adopting a Vulnerability Approach.* World Fire Stat. Bull. Valéria Pacella.
3. Pastor, E., Zárate, L., Planas, E., Arnaldos, J. (2003). Mathematical models and calculation systems for the study of wildland fire behaviour. *Progress in Energy and Combustion Science, 29 (2),* 139–153. doi: https://doi.org/10.1016/s0360-1285(03)00017-0
4. Masellis, M., Ferrara, M. M., Gunn, S. W. A. (1999). Fire disaster and burn disaster: Planning and management. *Annals of Burns and Fire Disasters, 12,* 67–76.
5. Jamesdaniel, S., Elhage, K. G., Rosati, R., Ghosh, S., Arnetz, B., Blessman, J. (2019). Tinnitus and Self-Perceived Hearing Handicap in Firefighters: A Cross-Sectional Study. *International Journal of Environmental Research and Public Health, 16 (20),* 3958. doi: https://doi.org/10.3390/ijerph16203958
6. Tedim, F., Leone, V., McCaffrey, S., McGee, T. K., Coughlan, M., Correia, F. J. M., Magalhães, C. G. (2020). Safety enhancement in extreme wildfire events. *Extreme Wildfire Events and Disasters.* Elsevier, 91–115. doi: https://doi.org/10.1016/b978-0-12-815721-3.00005-9
7. Heydari, A., Ostadtaghizadeh, A., Ardalan, A., Ebadi, A., Mohammadfam, I., Khorasani-Zavareh, D. (2022). Exploring the criteria and factors affecting firefighters' resilience: A qualitative study. *Chinese Journal of Traumatology, 25 (2),* 107–114. doi: https://doi.org/10.1016/j.cjtee.2021.06.001
8. Yoon, J.-H., Kim, Y.-K., Kim, K. S., Ahn, Y.-S. (2016). Characteristics of Workplace Injuries among Nineteen Thousand Korean Firefighters. *Journal of Korean Medical Science, 31 (10),* 1546. doi: https://doi.org/10.3346/jkms.2016.31.10.1546
9. Khoon, T. N., Sebastian, P., Saman, A. B. S. (2012). Autonomous Fire Fighting Mobile Platform. *Procedia Engineering, 41,* 1145–1153. doi: https://doi.org/10.1016/j.proeng.2012.07.294
10. Kim, J.-H., Starr, J. W., Lattimer, B. Y. (2014). Firefighting Robot Stereo Infrared Vision and Radar Sensor Fusion for Imaging through Smoke. *Fire Technology, 51 (4),* 823–845. doi: https://doi.org/10.1007/s10694-014-0413-6
11. Sivakumar, E. V., Manoj, P., Pumithadevi, S., Sylvia, S. S., Thangaraj, M. (2016). Voice Controlled Intellegent Fire Extinguisher Robot. *International Journal for Science and Advance Research in Technology, 2,* 65–67.
12. Sonal, M., Bharat, M., Saraswati, S., Bansude, V. U. (2017). Fire Fighting Robot. *International Research Journal of Engineering and Technology, 4,* 136–138.
13. Vijayalakshmi, B., Vasanth, A., Kumar, S. V., Raj, S. N. (2017). Autonomous Fire Fighting Robot With Self Power Management. *International Journal of Scientific Engineering and Research, 8,* 234–237.
14. Sucuoglu, H. S., Bogrekci, I., Demircioglu, P. (2018). Development of Mobile Robot with Sensor Fusion Fire Detection Unit. *IFAC-PapersOnLine, 51 (30),* 430–435. doi: https://doi.org/10.1016/j.ifacol.2018.11.324
15. Aliff, M., Samsiah, N., Yusof, M., Zainal, A. (2019). Development of Fire Fighting Robot (QRob). *International Journal of Advanced Computer Science and Applications, 10 (1).* doi: https://doi.org/10.14569/ijacsa.2019.0100118
16. Chandra, P. S., Revathi, V., Sireesha, A., Kumar, N. S. (2019). Development of DTMF Centred Remotely Located Fire Extinguishing Robot. *International Journal of Innovative Technology and Exploring Engineering, 9 (1),* 39–43. doi: https://doi.org/10.35940/ijitee.a3902.119119
17. Dhiman, A., Shah, N., Adhikari, P., Kumbhar, S., Dhanjal, I., Mehendale, N. (2020). Fire Fighter Robot with Deep Learning and Machine Vision. *SSRN Electronic Journal.* doi: https://doi.org/10.2139/ssrn.3633609
18. Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O. et al. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data, 8 (1).* doi: https://doi.org/10.1186/s40537-021-00444-8
19. Liu, H., Lang, B. (2019). Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey. *Applied Sciences, 9 (20),* 4396. doi: https://doi.org/10.3390/app9204396
20. Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. *Journal of Big Data, 2 (1).* doi: https://doi.org/10.1186/s40537-014-0007-7
21. Shen, C. (2018). A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists. *Water Resources Research, 54 (11),* 8558–8593. doi: https://doi.org/10.1029/2018wr022643
22. Atitallah, S. B., Driss, M., Boulila, W., Ghézala, H. B. (2020). Leveraging Deep Learning and IoT big data analytics to support the smart cities development: Review and future directions. *Computer Science Review, 38,* 100303. doi: https://doi.org/10.1016/j.cosrev.2020.100303
23. Wang, W., Liang, D., Chen, Q., Iwamoto, Y., Han, X.-H., Zhang, Q. et al. (2019). Medical Image Classification Using Deep Learning. *Deep Learning in Healthcare.* Springer, 33–51. doi: https://doi.org/10.1007/978-3-030-32606-7_3

24. Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., Swami, A. (2016). The Limitations of Deep Learning in Adversarial Settings. *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, 372–387. doi: https://doi.org/10.1109/eurosp.2016.36
25. Shorten, C., Khoshgoftaar, T. M., Furht, B. (2021). Deep Learning applications for COVID-19. *Journal of Big Data, 8 (1)*. doi: https://doi.org/10.1186/s40537-020-00392-9
26. Udendhran, R., Balamurugan, M., Suresh, A., Varatharajan, R. (2020). Enhancing image processing architecture using deep learning for embedded vision systems. *Microprocessors and Microsystems, 76,* 103094. doi: https://doi.org/10.1016/j.micpro.2020.103094
27. Peng, X., Zhang, X., Li, Y., Liu, B. (2020). Research on image feature extraction and retrieval algorithms based on convolutional neural network. *Journal of Visual Communication and Image Representation, 69,* 102705. doi: https://doi.org/10.1016/j.jvcir.2019.102705
28. Ramprasath, M., Anand, M. V., Hariharan, S. (2018). Image classification using convolutional neural networks. *International Journal of Pure and Applied Mathematics, 119 (17),* 1307–1319.
29. Liang, G., Hong, H., Xie, W., Zheng, L. (2018). Combining Convolutional Neural Network With Recursive Neural Network for Blood Cell Image Classification. *IEEE Access, 6,* 36188–36197. doi: https://doi.org/10.1109/access.2018.2846685
30. Albawi, S., Mohammed, T. A., Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*. doi: https://doi.org/10.1109/icengtechnol.2017.8308186
31. Ajit, A., Acharya, K., Samanta, A. (2020). A Review of Convolutional Neural Networks. *2020 International Conference on Emerging Trends in Information Technology and Engineering (Ic-ETITE)*. doi: https://doi.org/10.1109/ic-etite47903.2020.049
32. Shaheen, F., Verma, B., Asafuddoula, Md. (2016). Impact of Automatic Feature Extraction in Deep Learning Architecture. *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. doi: https://doi.org/10.1109/dicta.2016.7797053
33. Elangovan, P., Nath, M. K. (2020). Glaucoma assessment from color fundus images using convolutional neural network. *International Journal of Imaging Systems and Technology, 31 (2),* 955–971. doi: https://doi.org/10.1002/ima.22494
34. Elkhayati, M., Elkettani, Y. (2022). UnCNN: A New Directed CNN Model for Isolated Arabic Handwritten Characters Recognition. *Arabian Journal for Science and Engineering, 47 (8),* 10667–10688. doi: https://doi.org/10.1007/s13369-022-06652-5
35. Li, Z., Ren, K., Jiang, X., Li, B., Zhang, H., Li, D. (2022). *Domain generalization using pretrained models without fine-tuning.* doi: https://doi.org/10.48550/arXiv.2203.04600
36. Mehrotra, R., Ansari, M. A., Agrawal, R., Anand, R. S. (2020). A Transfer Learning approach for AI-based classification of brain tumors. *Machine Learning with Applications, 2,* 100003. doi: https://doi.org/10.1016/j.mlwa.2020.100003
37. Yao, Y., Doretto, G. (2010). Boosting for transfer learning with multiple sources. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1855–1862. doi: https://doi.org/10.1109/cvpr.2010.5539857
38. Bouhamed, O., Ghazzai, H., Besbes, H., Massoud, Y. (2020). Autonomous UAV Navigation: A DDPG-Based Deep Reinforcement Learning Approach. *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*. doi: https://doi.org/10.1109/iscas45731.2020.9181245
39. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T. et al. (2017). *Mobilenets: Efficient convolutional neural networks for mobile vision applications*. doi: https://doi.org/10.48550/arXiv.1704.04861

✉***Amin Saif,*** *Associate Professor, Department of Power Systems and Complexes, Academy of Sciences of the Republic of Uzbekistan, Tashkent, Uzbekistan; Department of Mechatronics and Robotics Engineering, Taiz University, Taiz, Yemen, e-mail: amye008@gmail.com, ORCID: https://orcid.org/0000-0001-5700-5644*

Gamal Muneer, *Assistant, Department of Mechatronics and Robotics Engineering, Taiz University, Taiz, Yemen, ORCID: https://orcid.org/0000-0002-5222-0924*

Yusuf Abdulrahman, *Assistant, Department of Electronic/Electrical Control Engineering, Aljanad University for Science and Technology, Taiz, Yemen, ORCID: https://orcid.org/0000-0001-6513-4550*

Hareth Abdulbaqi, *Postgraduate Student, Department of Mechatronics and Robotics Engineering, Taiz University, Taiz, Yemen, ORCID: https://orcid.org/0000-0002-5750-8935*

Aiman Abdullah, *Postgraduate Student, Department of Mechatronics and Robotics Engineering, Taiz University, Taiz, Yemen, ORCID: https://orcid.org/0000-0001-6356-344X*

Abdullah Ali, *Postgraduate Student, Department of Mechatronics and Robotics Engineering, Taiz University, Taiz, Yemen, ORCID: https://orcid.org/0000-0003-3559-7348*

Abduljalil Derhim, *Postgraduate Student, Department of Mechatronics and Robotics Engineering, Taiz University, Taiz, Yemen, ORCID: https://orcid.org/0000-0001-7877-854X*

✉*Corresponding author*