International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-9 Issue-1, October, 2019

Data -Enhanced Convolution Neural Networks for Wall Following Robot Navigation

Sandip Kumar Singh

Abstract*: Machine learning has been used for solving the Robot Navigation Task through the wall-following control. The wall-following control involves the movement of the robot in some directed direction maintaining a constant distance from a given wall. The path of the movement of robot is measured by ultrasonic sensors. Many machine learning methods have been used for this problem, as classifiers, but Convolution Neural Networks (CNN) outperforms them all with almost 98% of accuracy. This study compared the performance of five classifiers SVC, MLR, ANN, CNN-1D, and CNN-2D, which play the part of controller in the navigation work. We have used the ultrasonic sensor data to understand the hidden pattern in the navigation work and classified the actions by robot in terms of different motions performed by robot in response to it.

The classification reports of CNN-2D and CNN-1D with Artificial Neural Networks (ANN) have also been presented in this paper. The smart Data-Enhancement used in proposed method significantly improves the classification performance of all classifiers, especially CNN.

Keywords: Convolution Neural Networks (CNN), Wall-following robot navigation, Multinomial logistic Regression (MLR), Support Vector Classifier (SVC)

I. INTRODUCTION

A robot is an automatically working machine and can do some task that human can do. A mobile robot is specially designed to operate in environments such as automated assembly halls, factories, or warehouses [2]. The wall-following robot navigation is the movement of robot keeping a fixed distance from wall. The path of the movement of the robot is measured by ultrasonic sensors. [11]. The ultrasonic sensors are used in many numbers to generate data for wall and path following robots. This data is used as a feature for pattern learning and further utilized to control the robot motion. The immediate response of the robot after learning the sensor data is called real time obstacle avoidance. The sensor data is converted to usable data with the help of complex algorithms. Juang et al. [1] proposed fuzzy controller (FC) based reinforcement ant optimized design method, and applied it to wheeled-mobile-robot wall-following control. Ando, Y presents a method for an autonomous mobile robot with a sonar-ring to follow walls [8]. The sonar-ring consists of multiple ultrasonic range sensors. The proposed wall-following algorithm makes a robot able to follow a wall in various shapes. Hsu, C. H., & Juang, C. F. propose evolutionary wall-following control of a mobile robot using an interval type-2 fuzzy controller (IT2FC) with species- differential -evolution-activated continuous ant colony optimization [4]. Das, A. K describes the method of real-time estimation on a car-like robot using a single omnidirectional camera as a sensor [7]. The concepts of car maneuvers, fuzzy logic control (FLC), and sensor-based behaviors are merged to implement the human-like driving skills by an autonomous car-like mobile robot (CLMR) by Li, T. H [6]. Millan et al. used machine learning along with asynchronous electro cephalogram in advanced robotics [9] for control of a mobile robot. Ge, S. S describes the problem of goals unreachable with obstacles nearby when using potential field methods for mobile robot path planning [10]. Negishi, Y [3] describes the navigation of a mobile robot in unknown tactic environments using an omnidirectional stereo and a laser range finder. Brooks et al. used a new design methodology for control systems, known as subsumption architecture. It decomposed the problems in task- achieving behaviors [5]. Classifying data is a prime task of machine learning. Support vector machines (SVM) are used as binary classifiers. They are used both as supervised learning methods and unsupervised learning methods. When the data is labelled, classification by SVM is called SVC that is support vector classifier. In case of unsupervised learning where data is unlabelled, it is clustered in two distinct groups with the help of SVM. The Multinomial Logistic Regression (MLR) is a further extension of Logistic regression, which is a binary classifier. MLR classifies multiclass data both linearly and nonlinearly. Primary Logistic regression classifier is useful in binary situations like on /off, fail /pass, live/ dead, etc., where classes are discretely separable. In such situations, the classes can be labelled between "0" or "1". Multinomial Logistic Regression is used for the dependent variables which are categorically equivalent. This categorical equivalence of dependent variables is also called nomiality. This multiclass Logistic Regression is also called Multinomial Logistic Regression. We propose, a CNN based classification solution of wall-following control of robot navigation with outstanding performance of 98% classification accuracy.

Revised Manuscript Received on October 30, 2019.
* Correspondence Author
Sandip Kumar Singh*, Department of Mechanical Engineering, V B S Purvanchal University Jaunpur (U.P.), India. Sandipkumarsingh25@gmail.com

II. THE BASIC CONCEPT OF CNN AND PROPOSED APPROACH

The convolution neural networks (CNN) are different from artificial neural networks (ANN). The ANN consists of one input layer and other output layer with one hidden layer in between. On other hand, in CNN there are convolution layers followed by sub- sampling layers in between the input and output layers [12-13]. The input to the convolution layer is given in form of an image of m ×m ×r size, where r represents the number of channels which is equal to 3 for RGB image. A CNN consists of some convolutional and subsampling layers which are followed by fully connected layers. There are k number of filters in the convolution layer. The size of these filters is m × n × q, where m is the dimension of image and n is smaller than m, k is number filters, r is the number of channels and q can be equal to r, or less than it. Each map is then sub-sampled typically max-pooling over p x p regions with p ranges between 2 to 5 for smaller and larger inputs respectively. Figure 3 represents the architecture of CNN with number of convolution and sub- sampling layers.

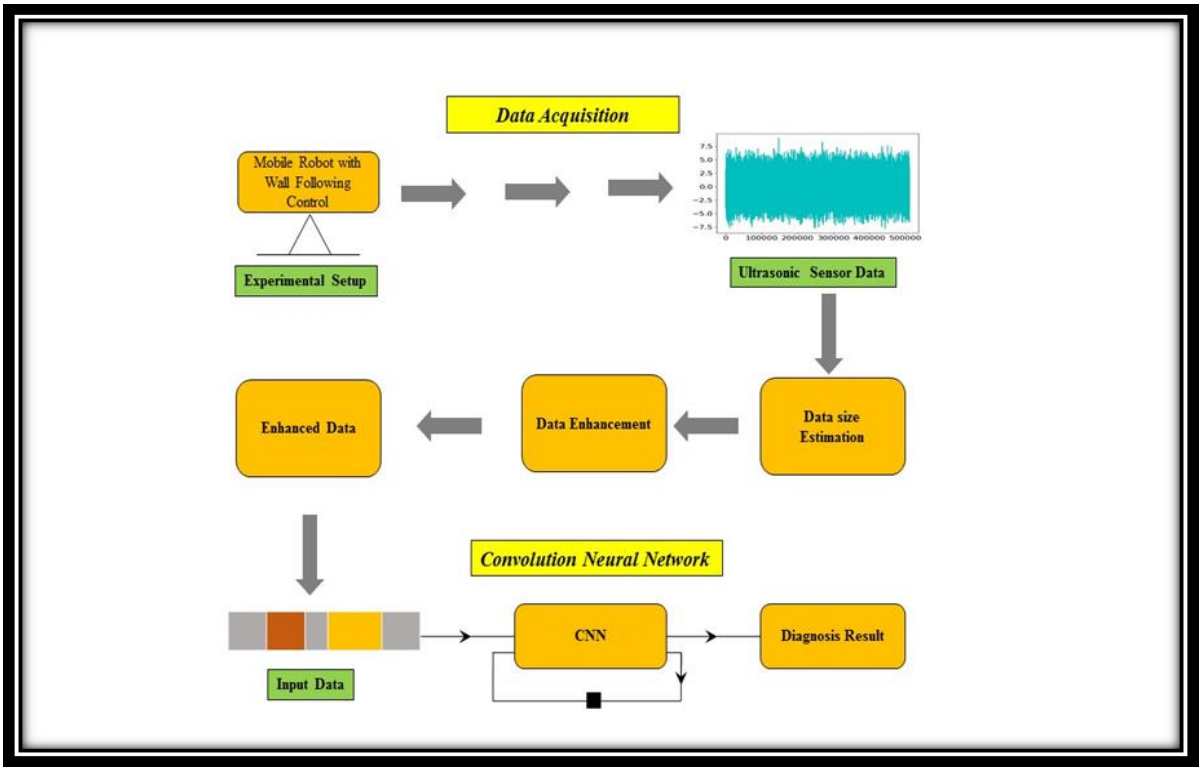

Fig. 1. Flow chart of Proposed Method

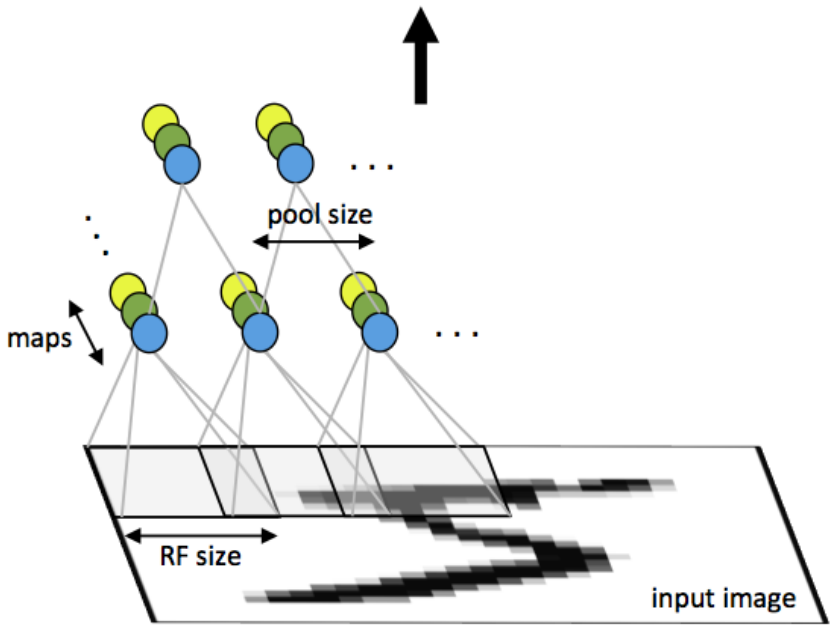

Fig. 2. The convolutional layer with pooling

Let δ^{l+1} be the error related term for the $(l+1)-$st layer in the network with a cost function $J(W,b;x,y)$, where (W,b) are the parameters and (x,y) are the data the $l-th$ layer is densely connected to the $(l+1)$-st layer, then the error for the l-th layer is computed as

$$\delta^{(l)} = ((w^{(l)})^T \delta^{(l+1)}).f'(z^{(l)})$$

Where $(f'(z^{(l)})$ is derivative of activation function and the gradients are

$$\nabla_{W^{(l)}} J(W,b;x,y) = \delta^{(l+1)}(a^{(l)})^T$$

If the l-th layer is a convolutional and subsampling layer, then the error is propagated through as

$$\delta_k^{(l)} = \textbf{unsample}\ ((W_k^{(l)})^T \delta_k^{(l+1)}).f'(z_k^{(l)})$$

Where k indexes the filter number, and $f'(z_k^{(l)})$ is the derivative of the activation function.

Lastly, to calculate the gradient w.r.t to the filter maps, $\delta_k^{(l)}$ the same way we flip the filters in convolution layer

$$\nabla_{W_k^{(l)}} J(W,b;x,y) = \sum_{i=1}^{m} a_i^{(l)}\, rot90(\delta_k^{(l+1)},2)$$

$$\nabla_{b_k^{(l)}} J(W,b;x,y) = \sum_{a,b} (\delta_k^{(l+1)})(\delta_k^{(l+1)})_{a,b}$$

Where, $\nabla_{W_k^{(l)}}$ is gradient of parameter W w.r.t. to $k-th$ filter, $\nabla_{b_k^{(l)}}$ is gradient of parameter b w.r.t. to $k-th$ filter, a^l is the input to the l-th layer, and a^l is the input image. The operation $a_i^{(l)} * (\delta_k^{(l+1)})$ is the "valid" convolution between $i-$th input in the $l-$th layer and the error w.r.t. the $k-$th filter.

In proposed CNN, we have used 5 layers which 600 hidden units. There are four response classes, accordingly output layers consists of 4 units. Hyperbolic tangent function has been used activation function. Decision has used zero bias 0. 9 momentum, and 0.0006 learning rate.

III. Data Description

The data was recorded @ 9 samples per second and generated a database with 5466 examples. The location of different items at the test place can be seen as in figure 5. The SCITOS G5 robot has been used for this purpose. The collection of data is done while the robot moves with the help of walls in a clockwise direction four times. The robot uses 24 ultrasound sensors to navigate, which are arranged circularly around its "waist." The numbering of the ultrasound sensors starts at the front of the robot and increases in the clockwise direction. The data contains the raw values of the measurements of all 24 ultrasound sensors and the corresponding class labels-- Move-forward, Slight-Right-Turn, Sharp-Right-Turn, Slight-Left-Turn. Sampling is done at a rate of 9 samples per second. This data has been taken from UCI data repository [12].

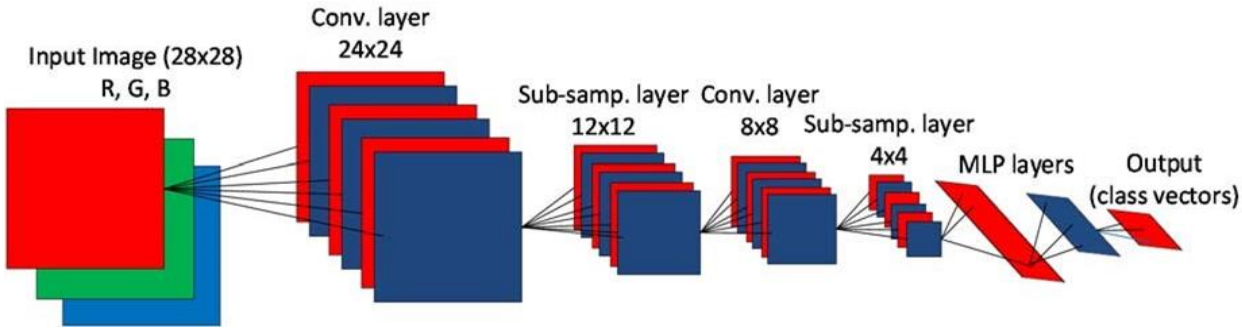

Fig. 3. 2D CNN configuration

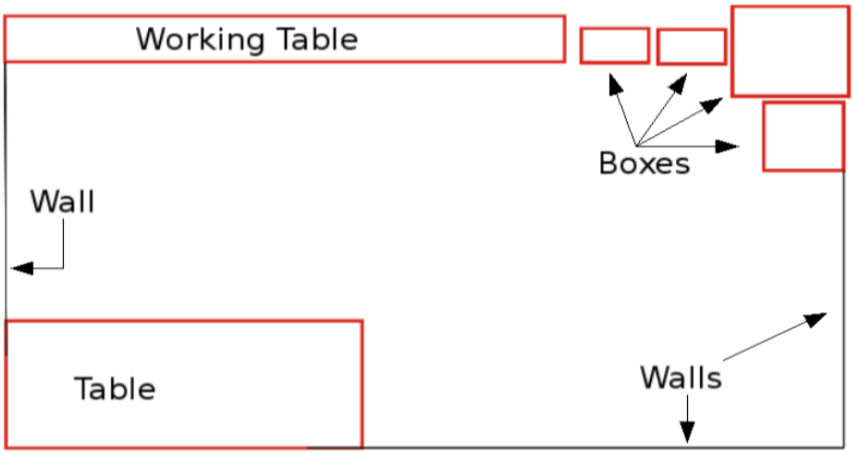

Fig.4. Sketch of the robot's navigation environment

IV. RESULT AND ANALYSIS

In Machine Learning prediction can be done in two ways, one is predicting positive results and the other is predicting negative results. The outcome of such predictions may be true positive and false positive, true negative and false negative results. Hence, total number of outcomes of any kind of predictions is of four types, which is true positive, true negative, false positive & false negative.

Precision is optimistically correct predictions (in fractions) with respect to total optimistic predictions.

Recall is optimistic predictions with respect to total number of correct predictions. It is also in fractions.

The occurrence of high-precision and high recall indicates that the predictor is capable of predicting correct and positive results simultaneously, which is sign of most effective predictor.

The combined effect of precision and recall is represented in terms of f1-score. Mathematically f1- score is the harmonic mean of precision and recall.

The precision-recall curve: The precision-recall curve depicts the relation between Precision and recall for different values. The high president implies that there is a low false-positive rate. The high recall means a low false-negative rate. When Precision and recall both are large, the area under curve is also very large, which is the most desired situation. Figs 5,6 & 7 show that the classifier is predicting correctly, and all predictions are positive.

The Precision-recall curves are generally useful in binary classification to understand the output of a machine learning classifier. The precision -recall technique is extended to a multiclass situation in our case.it is achieved by binarizing the output of result.

TABLE I. Classification Report of ANN

ANN				
Class/label	*precision*	*recall*	*f1-score*	*support*
Move-forward 0	0.95	0.96	0.95	1725
Slight-Right-Turn 1	0.95	0.88	0.91	535
Sharp-Right-Turn 2	0.97	0.98	0.97	1709
Slight-Left-Turn 3	1	0.96	0.98	255
Avg/Tot	0.96	0.96	0.96	4224
Confusion Matrix				
Class	0	1	2	3
Move-forward 0	1655	22	48	0
Slight-Right-Turn 1	57	472	6	0
Sharp-Right-Turn 2	29	3	1677	0
Slight-Left-Turn 3	8	0	3	244

TABLE.II Classification Report Of CNN-1D

CNN-1D				
Class/label	*precision*	*recall*	*f1-score*	*support*
Move-forward 0	0.97	0.96	0.96	1725
Slight-Right-Turn 1	0.94	0.92	0.93	535
Sharp-Right-Turn 2	0.97	0.99	0.98	1709
Slight-Left-Turn 3	0.99	0.99	0.99	255
Avg/ total	0.97	0.97	0.97	4224
Confusion Matrix				
Class	0	1	2	3
Move-forward 0	1655	27	43	0
Slight-Right-Turn 1	34	497	4	0
Sharp-Right-Turn 2	14	5	1690	0
Slight-Left-Turn 3	0	3	0	252

TABLE III. Classification Report of CNN-2D

CNN-2D				
Class/label	*precision*	*recall*	*f1-score*	*support*
Move-forward 0	0.97	0.96	0.97	1725
Slight-Right-Turn 1	0.93	0.93	0.93	535
Sharp-Right-Turn2	0.97	0.99	0.98	1709
Slight-Left-Turn 3	1	0.99	0.99	255
Avg/ total	0.97	0.97	0.97	4224
Confusion Matrix				
Class	0	1	2	3
Move-forward 0	1651	28	44	2
Slight-Right-Turn 1	39	491	5	0
Sharp-Right-Turn 2	11	4	1694	0
Slight-Left-Turn 3	2	0	0	253

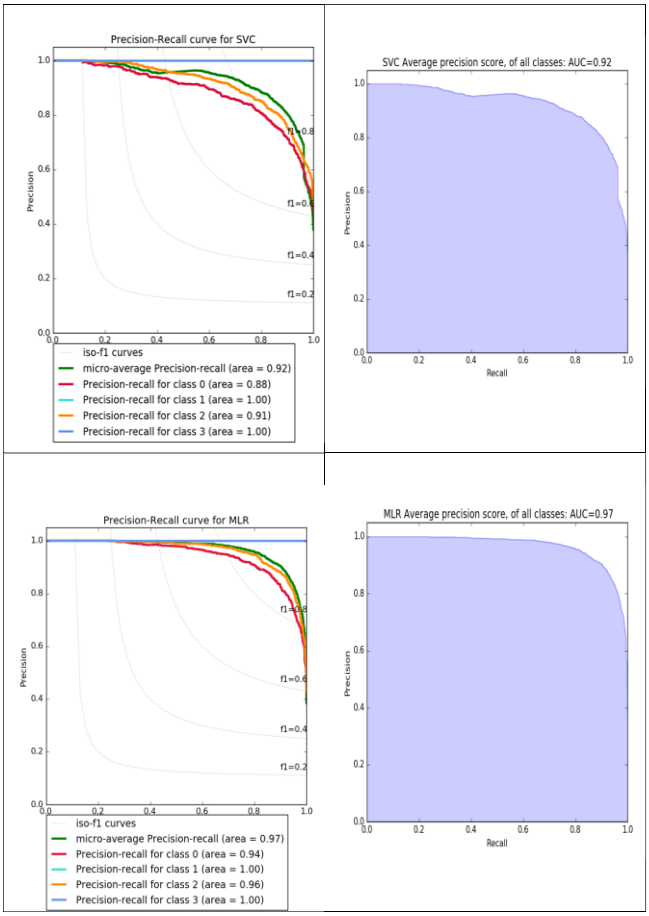

Fig.5. Precision-Recall curves for SVC and MLR

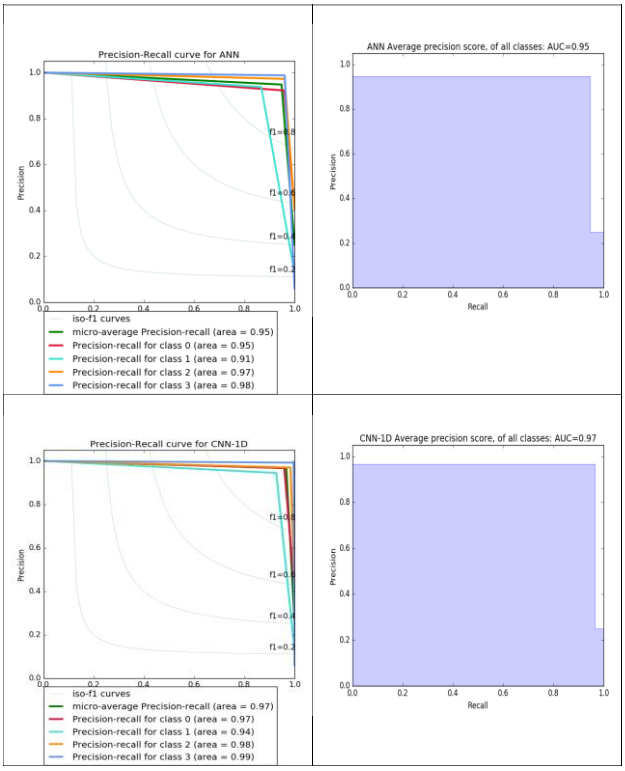

Fig.6. Precision-Recall curves for ANN and CNN-1D

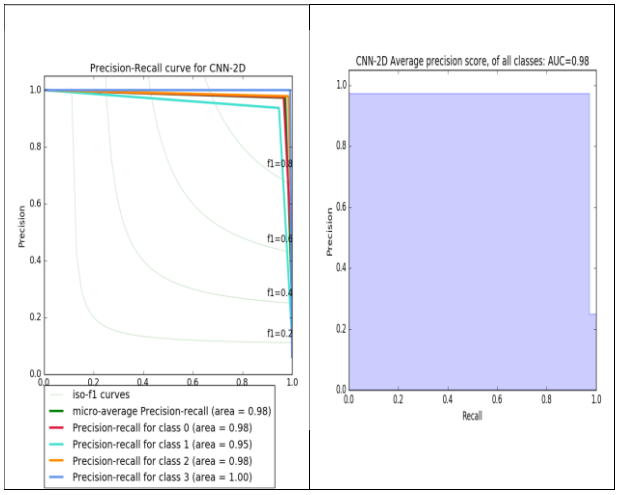

Fig.7. Precision-Recall curves for CNN-2D

V. CONCLUSION

Figures(5, 6, and 7) along with Tables (I,II, and III) represent the performance of 5 classifiers for wall following problem of robot navigation in the form of. Precision-recall curve and Confusion matrix. The Precision -recall curve and Confusion matrix are well-established measures of the performance of machine learning classifiers. Figure 6 shows that the Support Vector Classifiers(SVC) and Multinomial Logistic Regression(MLR) have 92 percent and 97 percent average area under curve (AUC), respectively, for Precision -recall curve. Figure 7 shows the performance of artificial neural network (ANN), which is slightly less effective than MLR.The two deep learning models CNN- 1D and CNN-2D perform excellently with the maximum area under curve for Precision- recall curve as 97 percent and 98 percent, respectively. It exhibits that Convolution Neural Networks(CNN) are very effective tools for wall-following control of robot navigation.The Confusion table of all classifiers show maximum error in class1 i.e.slight right turn movement,out of four motions of robot.Class3 i.e.slight left turn, is very well-identified by all classifiers.We also notice that proper data enhancement improves the classification efficiency of all classifiers particularly in those cases where data size is not sufficiently large.

REFERENCES:

1. Juang, C. F., & Hsu, C. H. (2009). Reinforcement ant optimized fuzzy controller for mobile-robot wall-following control. *IEEE Transactions on Industrial Electronics*, *56*(10), 3931-3940.
2. Roslinamaria, A. R. (2005). Wall Following Mobile Robot.
3. Negishi, Y., Miura, J., & Shirai, Y. (2004, September). Mobile robot navigation in unknown environments using omnidirectional stereo and laser range finder. In *Intelligent Robots and Systems, 2004. (IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on* (Vol. 3, pp. 2737-2742). IEEE.
4. Hsu, C. H., & Juang, C. F. (2013). Evolutionary robot wall-following control using a type-2 fuzzy controller with species-DE-activated continuous ACO. *IEEE Transactions on Fuzzy Systems*, *21*(1), 100-112.
5. Brooks, R. A., & Connell, J. H. (1987). Asynchronous distributed control system for a mobile robot. *Mobile Robots I*, *727*, 77-85.
6. Li, T. H., Chang, S. J., & Chen, Y. X. (2003). Implementation of human-like driving skills by autonomous fuzzy behavior control on an FPGA-based car-like mobile robot. *IEEE Transactions on Industrial Electronics*, *50*(5), 867-880.
7. Das, A. K., Fierro, R., Kumar, V., Southall, B., Spletzer, J., & Taylor, C. J. (2001). Real-time vision-based control of a nonholonomic mobile robot. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on* (Vol. 2, pp. 1714-1719). IEEE.
8. Ando, Y., & Yuta, S. I. (1995, May). Following a wall by an autonomous mobile robot with a sonar-ring. In *Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on* (Vol. 3, pp. 2599-2606). IEEE.
9. Millan, J. R., Renkens, F., Mourino, J., & Gerstner, W. (2004). Noninvasive brain-actuated control of a mobile robot by human EEG. *IEEE Transactions on Biomedical Engineering*, *51*(6), 1026-1033.
10. Ge, S. S., & Cui, Y. J. (2000). New potential functions for mobile robot path planning. *IEEE Transactions on robotics and automation*, *16*(5), 615-620.
11. Van Turennout, P., Honderd, G., & Van Schelven, L. J. (1992, May). Wall-following control of a mobile robot. In *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on* (pp. 280-285). IEEE.
12. Freire, A. L., Barreto, G. A., Veloso, M., & Varela, A. T. (2009, October). Short-term memory mechanisms in neural network learning of robot navigation tasks: A case study. In *Robotics Symposium (LARS), 2009 6th Latin American* (pp. 1-6). IEEE.
13. Singh, S. K., Kumar, S., & Dwivedi, J. P. (2017). Compound fault prediction of rolling bearing using multimedia data. *Multimedia Tools and Applications*, 1-18.

AUTHOR'S PROFILE

Sandip Kumar Singh is presently employed as an Associate Professor in the Department of Mechanical Engineering at V B S Purvanchal Universit,y Jaunpur (U.P.), India. He completed his B.Tech degree from Kamla NehruInstitute of Technology(KNIT) Sultanpur(U.P.), and did M.Tech. from National Institute of Technology (N I T) Kurukshetra. He has done Ph.D. from Indian Institute of Technology (IIT BHU) Varanasi. His area of interest is Machine Learning and Structural Health Monitoring.