University of Baghdad
College of Engineering
JE
JOURNAL OF ENGINEERING

Journal of Engineering

journal homepage: www.jcoeng.edu.iq

Volume 30 Number 4 April 2024

Single Spike Neural Network Model for Superficial Environment Classification for Mobile Robot Navigation

Zhraa Issam Ibrahim[1,*], Nadia Adnan Shiltagh Al-Jamali[2]

Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq
zahraa.ibrahim2105m@coeng.uobaghdad.edu.iq[1], nadia.aljamali@coeng.uobaghdad.edu.iq[2]

ABSTRACT

In this work, a single Semi-Recurrent Spike Neural Network (SRSNN) supervised learning method based on time coding is proposed to classify visual terrain encountered by the mobile robot. To this end, the features are extracted using the Local Binary Pattern (LBP) method. Then, the SRSNN is trained to classify multi-class of terrains. This training is used to classify six classes of terrain: hydrop, gravel, asphalt, grass, mud, and sand. The proposed training algorithm is based on adaptive synaptic weights that reach the threshold value. The feature extracted method and SRSNN form the proposed Intelligent structure. This structure effectively evaluates the accuracy, precision, recall, and F1 score. The simulation results achieve good performance in minimizing the mean square error in the training phase and maximizing the overall accuracy to 87.22%, especially in dangerous terrain (i.e. hydrop). The effectiveness of the proposed model is proved by the efficiency of the training algorithm which can learn fast with accurate results.

Keywords: Single Spike Neural Network(SSNN), Terrain Classification, Mobile Robots.

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2024.04.08

Article received: 05/06/2023
Article accepted: 21/10/2023
Article published: 01/04/2024

نموذج شبكة عصبية أحادية سبايك لتصنيف البيئة السطحية للملاحة الروبوتية المتنقلة

زهراء عصام ابراهيم*، نادية عدنان شلتاغ الجمالي

قسم هندسة الحاسبات، كلية الهندسة، جامعة بغداد، بغداد، العراق

الخلاصة

في هذا البحث ، تم اقتراح طريقة تعلم تحت إشراف الشبكة العصبية احاديه الاشواك ذات الارجاع الجزئي (SRSNN) تعتمد على الترميز الزمني لتصنيف التضاريس المرئية التي يواجهها الروبوت المتحرك. تحقيقا لهذه الغاية ، أولاً ، يتم استخراج الميزات باستخدام طريقة النمط الثنائي المحلي (LBP). بعد ذلك ، يتم تدريب SRSNN على تصنيف التضاريس الى فئات متعددة. يستخدم هذا لتصنيف ست فئات من التضاريس: الماء ، والحصى ، والأسفلت ، والعشب ، والطين ، والرمل. تعتمد خوارزمية التدريب المقترحة على الأوزان التكيفية للمشابك التي تصل إلى قيمة العتبة. تشكل طريقة استخراج الميزات مع SRSNN الهيكل الذكي المقترح. يوفر هذا الهيكل فعالية جيدة في تقييم الدقة(مدى قرب القيمة من قيمتها الحقيقية) والدقة(مدى تكرار القياس) والاسترجاع ودرجة F1. حققت نتائج المحاكاة أداءً جيدًا من حيث تقليل متوسط الخطأ التربيعي في مرحلة التدريب وتعظيم الدقة الكلية الى 87.22% ، لا سيما في التضاريس الخطرة (مثل الماء). تم إثبات فعالية النموذج المقترح من خلال كفاءة خوارزمية التدريب التي يمكن أن تتعلم بسرعة مع نتائج دقيقة.

الكلمات المفتاحية: شبكة عصبية احاديه الاشواك, تصنيف التضاريس, الروبوتات المتنقلة.

1. INTRODUCTION

These days robot plays important roles in human life. It is used in different fields such as military, medical services, and especially in dangerous places, including working with equipment, mining, disarming bombs, and investigating shipwrecks **(Al-Araji et al., 2019; Al-Araji and Ibraheem, 2019; Atiyah and Hassan, 2023; Jawad and Hadi, 2019).** Therefore, the mobile robot must know its surrounding outdoor environment to perform its tasks safely. Autonomous ground robots confront many defiances in outdoor environments **(DuPont et al., 2008; Yu et al., 2021)**, especially in different types of terrain that impact the robot's power and motion **(Zou et al., 2020)**. For example, grassland and sand will cause energy exhaustion in mobile robots **(Wang et al., 2022; Hanson et al., 2022)**. Most motility control algorithms presume that the robot is moveable on solid ground. Therefore, only the shape of the terrain is considered. However, some of the terrain has the same shape but different materials. Thus, the type of material directly impacts the robot's motion **(Kozlowski and Walas, 2018; Zhang et al., 2016)**. Therefore, determining the kind of terrain confronted by mobile robots will facilitate navigation and decide the optimal next move. Hence, terrain classification is important for Autonomous ground robots to perform their tasks safely.

In recent years, many studies have been conducted on terrain classification, which can be classified into Proprioceptive and Exteroceptive methods. The former classifies terrain

during the robot traversable based on information resulting from the interaction between the wheel and the ground **(Zürn et al., 2020)**. However, this approach requires a vehicle platform that is more forgiving or more reactive to failures such as rollovers or crashes **(Wu et al., 2019)**. The latter method identifies forthcoming terrain (i.e. before the robot is traversable). It can split into geometry-based and appearance-based. Counting just on geometric features can lead to some ambiguity. For example, long grass and short fences have similar geometric features. However, usually, appearance-based visually appreciate a discrete set of terrain classes **(Papadakis, 2013)**. A spiking neural network provides time-based encoding, improving energy efficiency **(Shiltagh and Abas, 2015; Miao et al., 2018; Wu et al., 2019; Lee et al., 2018)**. A spiking neural network is used to classify various terrain types based on vision data. Many types of research have been done to tackle the terrain classification problem. These studies differ in terms of the sensors used and methods of classification. Most of them used vision-based sensors, as illustrated below:

(Wang et al., 2022) proposed a hybrid method (Convolutional Neural Network (CNN) and Support Vector Machine(SVM)) for terrain classification encountered by mobile robots. The result of the classification accuracy of the proposed method compared with the InpectionV3 and the CNN. The proposed method achieved high classification accuracy, especially for dangerous terrain types (i.e. hydrop), while maintaining high efficiency. **(Chen et al., 2021)** developed hybrid models for terrain classification tasks with vision-based and proprioception-based. One dimension Convolution Neural Network (CNN) model for proprioception net and a pre-trained (CNN) model for vision net were proposed. Fusion net combines vision net and proprioception net with two schemes: Decision-Level and Feature map-Level. The result was that the Decision-Level fusion model achieved the highest average accuracy of 96.40% of testing on four sets(test set, dark set, subset, and fog set). **(Vulpi et al., 2021)** proposed deep learning(LSTM, C-LSTM, and CNN) algorithm for tackling terrain classification problems based on proprioceptive sensing. Compared with SVM, the result was that the proposed algorithm achieved better accuracy for CNN, which was 91.5%. **(Zou et al., 2020)** developed a terrain classification algorithm(r-SNN) to classify three terrain types (i.e. grass, dirt, and road), including recurrent and supervised layers. The error rate was computed using three models for terrain classification (i.e. images only, sensors only, and both images and sensors) for this approach. It was compared with SVM and 3L- logistic regression. The developed approach achieved a better classification accuracy of over 90% with an error rate of 3.5% for images with sensors model. **(Kozlowski and Walas, 2018)** proposed Deep Neural Network(DNN) architecture for terrain classification problems. RGB-D sensor provides vision data, including RGB images, depth data, and infrared images. The proposed method achieved a 98.41% average classification accuracy on a test set using RGB data only and a 98.99% average classification accuracy on a test set using RGB, depth, and infrared data. **(Schilling et al., 2017)**presented a multi-sensor terrain classification system based on geometric and visual features. Random forest was used to classify the terrain into three classes (i.e. risky, safe, and obstacle). The classification performance is better when these classes train the classifier on combined features.

The previously published work could implement terrain classification tasks based on neural networks that require large amounts of energy. This work proposes the terrain classification method based on vision data to classify six types of terrain encountered by mobile robots. A modified training algorithm based on time encoding called a single Semi-Recurrent Spike Neural Network (SRSNN) is proposed to provide low power consumption, computation speed, and energy efficiency.

2. THE PROPOSED MODEL OF TERRAIN CLASSIFICATION

Considering the merit of tackling terrain classification problems with vision-based methods and the vigorous performance of spiking neural networks, it is required to design a method to apply spiking neural networks to vision-based terrain classification for mobile robots. This is important to achieve high classification accuracy and efficiency. As shown in **Fig. 1**, the proposed model contains feature extraction and classification. The former is performed using the Local Binary Pattern (LBP) method for extracting texture features **(Karis et al., 2016; Singh et al., 2018; Humeau-Heurtier, 2019)**.

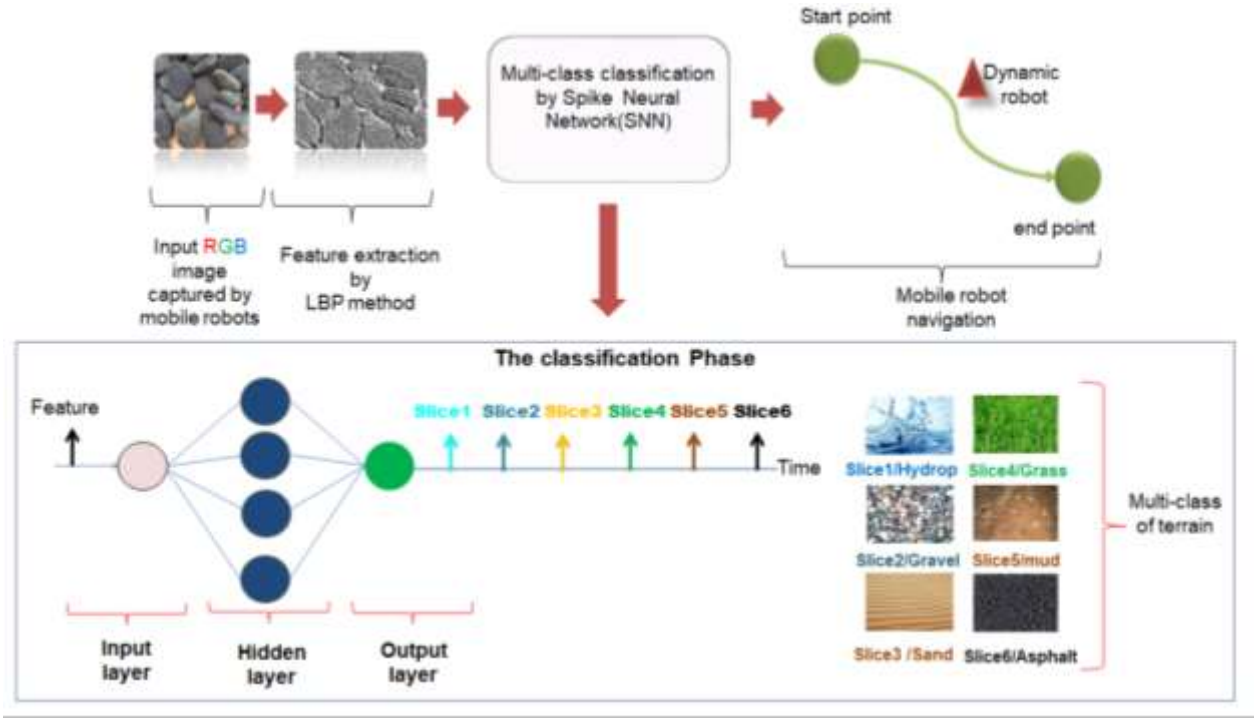

Figure 1. The proposed system model for multi-classes terrain classification.

However, the outdoor terrain's visual appearance is mutable during the days and seasons. Hence, the texture feature is used for discriminating between terrain classes. The result from the LBP method is a feature vector fed to the classification part. In this experiment, the SRSNN classifies the terrain into six types: hydrop, gravel, asphalt, grass, mud, and sand. Then, the dynamic robot will move according to this classification result.

3. SPIKING NEURAL NETWORK

Many studies have explored Spiking Neural Networks (SNN) to create low-power neuromorphic devices **(Kim et al., 2018; Hu et al., 2021)**. Neuromorphic architectures have the potency to dominate outdoor robotics under severe power curbs. Contrary to orthodox von Neumann architecture, neuromorphic architecture exhausts less power because of the massive concurrency and event-driven processing **(Indiveri et al., 2011; Lin et al., 2021)**.

3.1 Semi-Recurrent Spike Neural Network Structure

The Semi-Recurrent Spike Neural Network (SRSNN) structure, as shown in **Fig. 2**, consists of three layers: (input, hidden, and output) layer. Each connection between two neurons incorporates synapses. Every synaptic has various delays and weights. Any spiking neuron includes three computation phases: The first phase involves adding up all of the neuron's inputs. The second phase comprises determining whether the membrane potential exceeds the threshold. Finally, the third stage represents emitting a spike and restoring the membrane potential to zero. In this paper, the SRSNN structure contains six neurons in the input layer to define the features vector with size 18 elements, 42 neurons in the hidden layer, and six neurons in the output layer to represent six classes.
Additionally, this structure contains feedback from each neuron in the output layer to all neurons in the hidden layer. With feedback, the input includes the previous output and the present input. Hence, the memory of the model will be enhanced. Thus enhancing the performance of the model and making it more accurate.

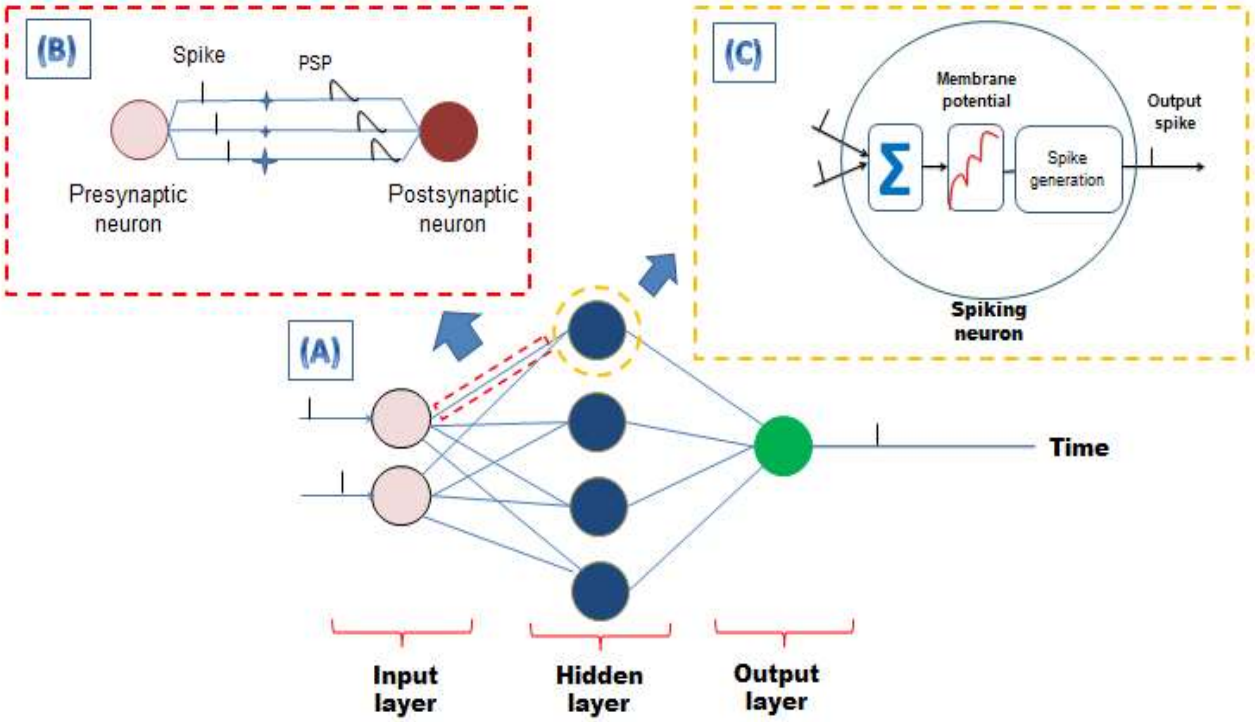

Figure 2. Structure of SNN: (A) Feed-forward of SNN. (B) The connection between two neurons. (C) Computational phases of the spiking neuron.

3.2 The Modified Training Algorithm

Spiking neural networks cope with pulse information instead of real data. Therefore, the first step for training is encoding data into spike times using the following formula **(Al-Yassari and Al-Jamali, 2023)**.

$$t_n^f = T_{max} - round\left(T_{min} + \frac{(P_{in}-P_{min})(T_{max}-T_{min})}{P_{max}-P_{min}}\right) \quad (1)$$

where T_{max} *and* T_{min} are the highest and smallest interval times, while the P_{min} *and* P_{max} are the smallest and largest extracted input, successively. P_{in} is the current real value of data. $round$ is a function that returns a number rounded to a certain number of digits.

The following action is the feed-forward phase, which starts from the hidden layer. All hidden neurons are examined to check if they have spiked or not. Each neuron is firing spike time only once throughout the time interval when its membrane potential transcends the threshold value ϑ, where membrane potential ($mp_h(t)$) in the hidden layer can be calculated with the added feedback of the output layer.

The feedback from the output layer to the hidden layer is added. Each neuron in the output layer is feedback to all neurons in the hidden layer. **Fig. 3** shows the sub-connection between two neurons with output feedback. It can be expressed by Eq. (2).

$$\text{FB} = \alpha \sum_{y=1}^{Y} \sum_{k=1}^{K} w_{hy}^{k} * o_y^s(t-1) \quad (2)$$

where α refers to the learning rate, Y is the number of neurons in the output layer, w_{hy}^{k} is the weight between hidden neuron (h) and output neuron (y), and $o_y^s(t-1)$ is the previous output spike of the output layer.

The membrane potential ($mp_h(t)$) is calculated by the following Eq. (3).

$$mp_h(t) = \sum_{n=1}^{N} \sum_{k=1}^{K} w_{nh}^{k} \varepsilon(t - t_n^f - d^k) + FB \quad (3)$$

where N is the count of presynaptic neurons, K is the index of synapses between two neurons, w_{nh}^{k} is the coefficient of the weight of synapse between presynaptic neuron (n) and post-synaptic neuron (h), t_n^f is the spike time of presynaptic neurons, d^k is the synapse's latency, and $\varepsilon(t)$ is the Spike Response Function(SRF). The SRF is a more biologically plausible model **(Yellakuor et al., 2020; Oniz et al., 2013).** It has various mathematical forms**.** The hyperbolic tangent function is used, as in Eq. (4).

$$\varepsilon(t) = tanh\left(\frac{t}{\tau}\right) \quad (4)$$

and its derivative, as below:

$$\frac{d\varepsilon}{dt} = \frac{1}{\tau}\left(1 - tanh^2\left(\frac{t}{\tau}\right)\right) \quad (5)$$

where τ is the time constant that determines the growth and decay of.

After all neurons in the hidden layer are checked, the same procedure is reiterated on the output layer. Finally, the output spike time information is converted to real information using Eq. (6).

$$RI\left(t_y^f\right) = \frac{\left(t_{max} - t_y^f - t_{min}\right) \times (P_{max} - P_{min})}{(t_{max} - t_{min})} + P_{min} \quad (6)$$

where $RI\left(t_y^f\right)$ is the real information of actual output spike time.

The Mean Square Error (MSE) calculates the overall error, as shown in Eq. (7).

$$MSE = 1/2 \sum_{y \epsilon Y} (t_y^f - t_y^d)^2 \quad (7)$$

where t_y^d indicates the training desired spike time for output neuron (y). The ensuing stage is back-propagation. This work uses the supervised learning back-propagation algorithm (SPIKE-PROP) to update each synapse's weight to minimize the MSE value, as in **(Thiruvarudchelvan et al., 2013)**. The weight of synapses between the output neuron and the hidden neuron is updated per Eq.s (8 to 10).

$$\delta_y = \frac{t_y^d - t_y^f}{\sum_{h=1}^{H}\sum_{k=1}^{K} w_{hy}^k \frac{\partial}{\partial t} y_h^k} \tag{8}$$

$$\Delta w_{hy}^k = \alpha . \delta_y y_h^k \tag{9}$$

$$w_{hy}^k(t+1) = w_{hy}^k(t) - \Delta w_{hy}^k \tag{10}$$

where δ_y points out the delta function of output layer neurons.
The weights of synapses between the hidden neuron and input neuron are updated following Eq.s (11, 12 and 13).

$$\delta_h = \frac{\sum_{h=1}^{H} \delta_y \sum_{k=1}^{K} w_{hy}^k(t) \frac{\partial}{\partial t} y_h^k}{\sum_{n=1}^{N}\sum_{k=1}^{K} w_{nh}^k(t) \frac{\partial}{\partial t} y_n^k} \tag{11}$$

$$\Delta w_{nh}^k = \alpha . \delta_h y_n^k \tag{12}$$

$$w_{nh}^k(t+1) = w_{nh}^k(t) - \Delta w_{nh}^k \tag{13}$$

where δ_h, N point out the delta function of hidden layer neurons and index of input layer neurons, respectively.

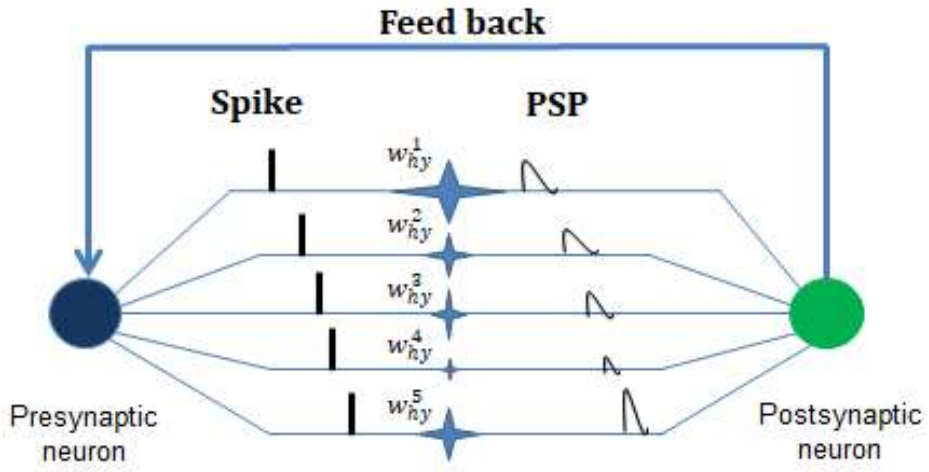

Figure 3. The synapses of one connection between hidden neuron and output neuron with output feedback.

The Pseudo code of the algorithm is used to represent the proposed model as shown in **Table 1**.

Table 1. Pseudo code of the proposed model.

Terrain classification proposed model in general shape
1: Start 2: S(*): Classification by the SRSNN 3: L(*): Feature extraction by LBP 4: E(*): Encoding the real value of Feature extraction into the spike 5: H: represent Hydrop class 6: Y: represent the other five classes 7: Input 8: I: Input image of terrain 9: Output 10: O: Output result of the classification 11: F← L(I) 12: * F is the texture feature vector in real number *\ 13: T← E(I) 14: * T is the texture feature vector in spike time *\ 15: R← S(T) 16: * R is the result of multi-class of SRSNN classifier, possible values of R is H or Y *\ 17: O← R 18: if O=H 19: * The robot doesn't move and goes back to step 1*\ 20: else 21: * The speed values are sent to each on-board motor for moving.*\ 22: End

4. SIMULATION RESULTS

In this section, the performance of the SRSNN classifier is evaluated. First, the SRSNN is implemented by using Spyder 2022v5. The collected dataset contains 42,000 images for six classes with (256x256) image size (https://www.pinterest.com/dsfafsh/textures/). Hence,70% of the dataset is used for training and 30% for testing. This ratio is adopted in the dataset to avoid overfitting and under fitting. A confusion matrix has been used for deriving four values to evaluate the trained model performance **(Nasser and Behadili, 2022; Abdulrezzak and Sabir, 2023; Soud and Al-Jamali, 2023)**, as given in **Table 2.**

Table 2. Confusion matrix structure

Real value	Predictive value		
		Positive	Negative
	Positive	TP	FN
	Negative	FP	TN

- **True Positive(TP):** gives the count of the number correctly classified for positive class(i.e. real value is positive class and the predictive value is positive class).
- **True Negative(TN):** gives the count of the number correctly classified for negative class(i.e. real value is negative class and the predictive value is negative class).
- **False Positive(FP):** represents the number of misclassification for a negative class(i.e.

real value is a negative class, and the result of predictive value is a positive class).

- **False Negative(FN):** represents the number of misclassification for positive class(i.e. real value is positive class, and the result of predictive value is a negative class).

Performance metrics are accuracy, precision, recall, and F1 scores, calculated based on the above-stated TP, TN, FP, and FN.

- Accuracy: is the ratio of the sample correctly classified to all samples. as shown in Eq. (14).

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} * 100\% \quad (14)$$

- Precision: represents the true positive value to the total number of positive values predicted. As shown in Eq. (15).

$$\text{Precision} = \frac{TP}{TP+FP} * 100\% \quad (15)$$

- Recall: is the measure of positive class prediction to all actual positive samples. The recall is defined in Eq. (16).

$$\text{Recall} = \frac{TP}{TP+FN} * 100\% \quad (16)$$

- F1-score is the harmonic mean between precision and recall. As expressed in Eq. (17), the larger value of this metric means the implemented model is more efficient.

$$\text{F1-score} = \frac{(2*Recall*Precision)}{Precision+Recall} \quad (17)$$

The parameters of each neuron used in the implementation of SRSNN are τ_s, α, and ϑ whose values are 10ms, 0.001, and 1v, successively.

The spike neural network is characterized by dealing with time(spike) to represent information. **Figure 4** displays the spike of the output neuron. As shown in the Figure, when the membrane potential exceeds the threshold value, the neuron fires spike at 6 ms. Then, the membrane potential resets to zero.

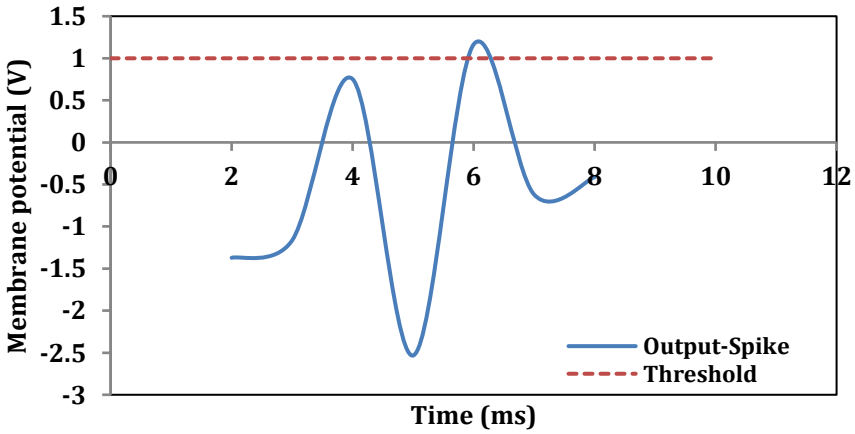

Figure 4. Spike of output neuron.

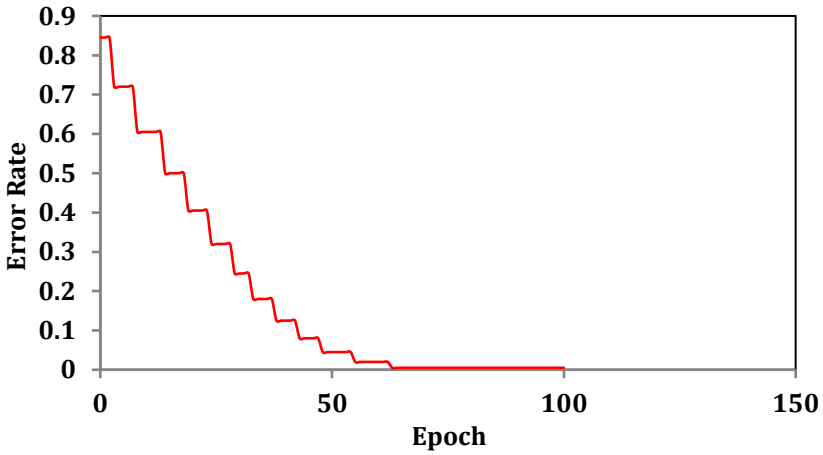

Figure 5. Error rate of SRSNN model.

Table 3. Performance metrics of the SRSNN model.

Performance metrics	The classes					
	Hydrop	Gravel	Grass	Asphalt	Mud	Sand
Accuracy %	97.532%	95.929%	95.357%	95.477%	95.556%	94.603%
Recall %	94.048%	89.762%	85.714%	88.095%	80.952%	84.762%
Precision %	91.351%	86.349%	86.331%	85.253%	91.398%	83.178%
F1-score	92.679	88.022	86.021	86.651	85.858	83.963

Figure 5 shows the error rate of the SRSNN model. The error rate value reaches to error goal at epoch 80. The performance metrics (accuracy, precision, recall, and F1-score) of the SRSNN learning algorithm are shown in **Table 3**, where **Fig. 6** represents the Accuracy of each class. It shows the classification accuracy of hydrop is higher than other terrains. This is because the misclassification of other terrains has increased, and the surface texture of hydrop terrain can be more discriminative.

However, relying just on the accuracy metric creates some ambiguity, giving no information about FP and FN. The recall metric is an essential metric for comprehending FNs, while the precision metric is for understanding FP. Therefore, the precision ratio and recall ratio are used.

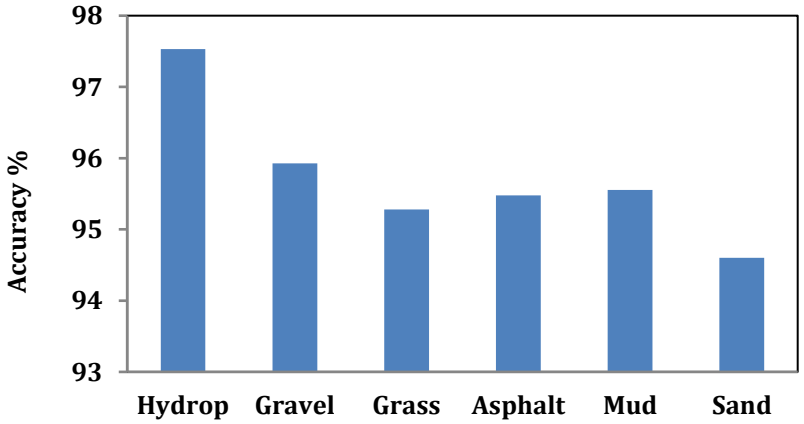

Figure 6. The Accuracy of the SRSNN model.

The importance of both FP and FN depends on the function of the classifier. In this paper, the significance of classes is various. The hydrop class is important due to its significant effects on mobile robots, leading to damage. In this case, the impact of FN appears because it shows the hydrop class's misclassification. When the recall ratio is high, this class can have less misclassification, see **Fig. 7**. However, the other classes have less importance compared with the significance of the hydrop class. The precision ratio of each class, where the low precision ratio means a high false positive for the class, is reported in **Fig. 8**. Finally, **Fig. 9** shows that the F1-score illustrates the equilibrium between recall and precision.

The overall accuracy of the model is compared with **(Wang et al., 2022)**, where the SRSNN achieves better performance in accuracy as compared with **(Wang et al., 2022)**, as shown in **Table 4**. This is because the intelligent structure used by the SRSNN makes the model more accurate. The spiking neural network also deals with spike time, unlike the traditional neural network, which deals with real data.

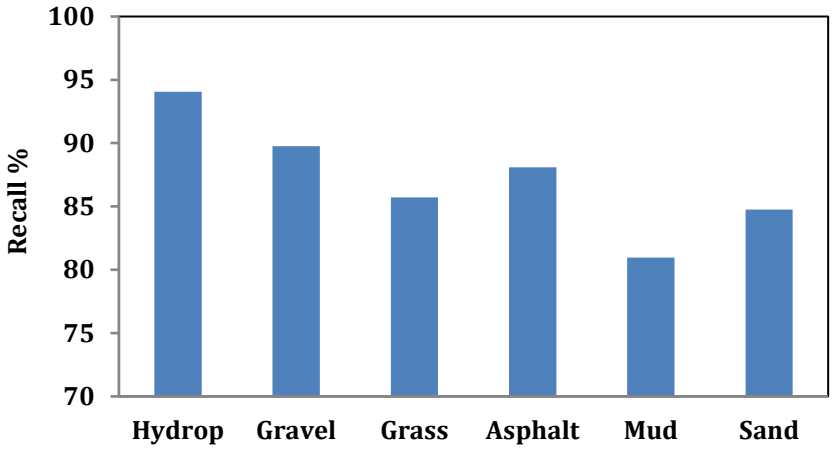

Figure 7. The Recall ratio of the SRSNN model.

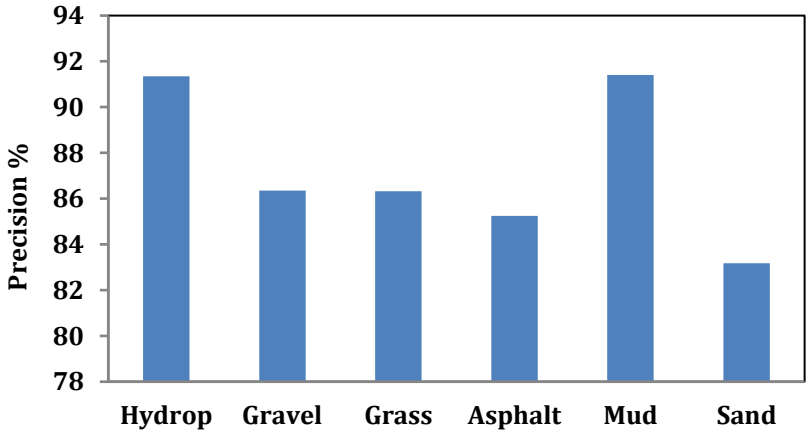

Figure 8. The precision ratio of the SRSNN model.

Table 4. The overall accuracy of the SRSNN model, and **(Wang et al., 2022)**.

Model	Year	Results	Summary
CNN+SVM	2022	82.90 %	Proposed hybrid method Convolutional Neural Network (CNN) and Support Vector Machine(SVM) to classify terrains encountered by mobile robots visually. CNN is used for multi-class of six-type terrain classification(hydrop, sand, mud, gravel, asphalt, grass) and SVM for two-class of one-type terrain classification (hydrop or other five types).
Our	2023	87.22 %	The SRSNN is used as a classifier to make the multi-class of terrain types (hydrop, sand, mud, gravel, asphalt, grass). The SRSNN provides energy efficiency, computation speed, and low power consumption.

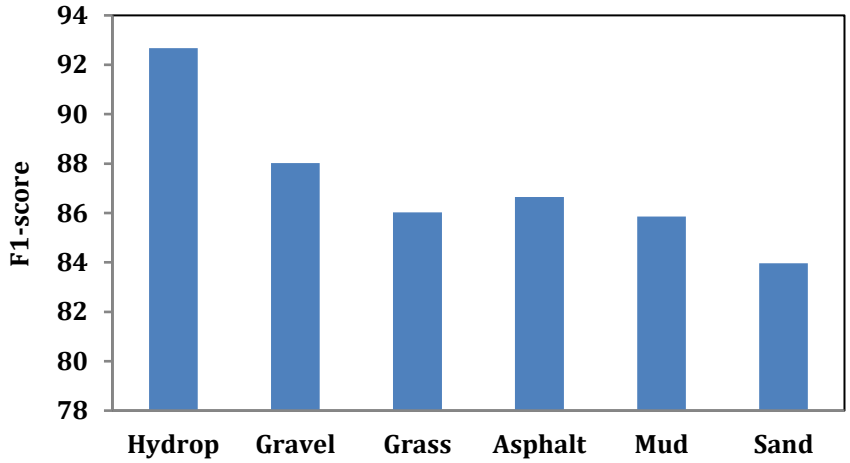

Figure 9. The F1-score of the SRSNN model.

5. CONCLUSIONS

This paper introduces a modified structure based on SNN with a training algorithm to solve the terrain classification problem based on vision data. In addition, this structure provides a powerful performance that allows presenting the SRSNN supervised learning method, where the SRSNN provides a multi-class terrain classification based on texture features. The features are extracted by using the LBP method. The modified SRSNN can minimize RMSE in a very short time. This is because in SRSNN, the neuron updates its weight only when the membrane potential exceeds the threshold value. The simulation results show that the SRSNN performs better in accuracy, precision, recall, and F1-score, especially in dangerous terrain (i.e. hydrop). This method serves the autonomous navigation of the robotic system, especially in path-planning robotic systems. In future work, the current researchers are going to fusion visual and geometric features instead of using only visual features.

NOMENCLATURE

Symbol	Description	Symbol	Description
α	learning rate	t_y^d	training desired spike time
FB	output feedback	t_y^f	actual output spike time
FP	False Positive	T_{max}	highest interval times

FN	False Negative	T_{min}	smallest interval times
H	count of hidden neurons	TN	True Negative
K	index of synapses between two neurons	TP	True Positive
mp_h	membrane potential of hidden layer	w_{nh}^k	coefficient of the weight of synapse between presynaptic neuron (n) and post-synaptic neuron (h)
MSE	mean square error	w_{hy}^k	weight between hidden neuron (h) and output neuron (y)
N	count of presynaptic neurons	Y	number of neurons in output layer
P_{in}	current real value of data	y_h^k	SRF for hidden neurons
P_{min}	smallest extracted input	y_n^k	SRF for input neurons
P_{max}	largest extracted input	δ_h	delta function of hidden layer neurons
RI	real information	δ_y	delta function of output layer neurons
$round$	a function that returns a number that has been rounded to a certain number of digits	$\varepsilon(t)$	spike response function (SRF)
t_n^f	spike times	$o_y^s(t-1)$	previous output spike of output layer

Acknowledgements

The authors would like to thank the University of Baghdad, College of Engineering, Computer Engineering Department for their support.

Credit authorship contribution statement

Zhraa Issam Ibrahim: Writing – original draft. Nadia Adnan Shiltagh Al-Jamali: Writing – review & editing, Validation, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

REFERENCES

Al-Araji, A.S., Ahmed, A.K., and Dagher, K.E., 2019. A cognition path planning with a nonlinear controller design for wheeled mobile robot based on an intelligent algorithm. *Journal of Engineering*, *25*(1), pp. 64-83. Doi:10.31026/j.eng.2019.01.06

Al-Araji, A.S., and Ibraheem, B.A., 2019. A comparative study of various intelligent optimization algorithms based on path planning and neural controller for mobile robot. *Journal of Engineering*, *25*(8), pp. 80-99. Doi:10.31026/j.eng.2019.08.06

Atiyah, H.A., and Hassan, M.Y., 2023. Outdoor Localization for a Mobile Robot under Different

Weather Conditions Using a Deep Learning Algorithm. *Journal Européen des Systèmes Automatisés*, 56(1), P. 1. Doi:10.18280/jesa.560101

Jawad, M.M., and Hadi, E.A., 2019. A Comparative study of various intelligent algorithms based path planning for Mobile Robots. *Journal of Engineering*, 25(6), pp. 83-100. Doi:10.31026/j.eng.2019.06.07

DuPont, E.M., Moore, C.A., and Roberts, R.G., 2008. Terrain classification for mobile robots traveling at various speeds: An eigenspace manifold approach. I*EEE International Conference on Robotics and Automation* (pp. 3284-3289). Doi:10.1109/ROBOT.2008.4543711

Yu, Z., Sadati, S.H., Wegiriya, H., Childs, P., and Nanayakkara, T., 2021. A method to use nonlinear dynamics in a whisker sensor for terrain identification by mobile robots. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8437-8443. IEEE. Doi:10.1109/IROS51168.2021.9636571

Zou, X., Hwu, T., Krichmar, J., and Neftci, E., 2020. Terrain classification with a reservoir-based network of spiking neurons. *IEEE International Symposium on Circuits and Systems (ISCAS)* (pp. 1-5). Doi:10.1109/ISCAS45731.2020.9180740

Wang, W., Zhang, B., Wu, K., Chepinskiy, S.A., Zhilenkov, A.A., Chernyi, S., and Krasnov, A.Y., 2022. A visual terrain classification method for mobile robots' navigation based on convolutional neural network and support vector machine. *Transactions of the Institute of Measurement and Control*, *44*(4), pp. 744-753. Doi:10.1177/0142331220987917

Hanson, N., Shaham, M., Erdoğmuş, D., and Padir, T., 2022. Vast: Visual and spectral terrain classification in unstructured multi-class environments. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3956-3963. Doi:10.1109/IROS47612.2022.9982078

Kozlowski, P., and Walas, K., 2018. Deep neural networks for terrain recognition task. *Baltic URSI Symposium (URSI),* pp. 283-286. Doi:10.23919/URSI.2018.8406736

Zhang, H., Dai, X., Sun, F., and Yuan, J., 2016. Terrain classification in field environment based on Random Forest for the mobile robot. *35th Chinese Control Conference (CCC),* pp. 6074-6079. Doi:10.1109/ChiCC.2016.7554310

Zürn, J., Burgard, W., and Valada, A., 2020. Self-supervised visual terrain classification from unsupervised acoustic feature learning. *IEEE Transactions on Robotics,* 37(2), pp. 466-481. Doi:10.1109/TRO.2020.3031214

Wu, H., Zhang, W., Li, B., Sun, Y., Duan, D., and Chen, P., 2019. Visual terrain classification methods for mobile robots using hybrid coding architecture. *IEEE 4th International Conference on Image, Vision and Computing (ICIVC)*, pp. 17-22. Doi:10.1109/ICIVC47709.2019.8981092

Papadakis, P., 2013. Terrain traversability analysis methods for unmanned ground vehicles: A survey. *Engineering Applications of Artificial Intelligence*, *26*(4), pp. 1373-1385. Doi:10.1016/j.engappai.2013.01.006

Shiltagh, N.A., and Abas, H.A., 2015. Spiking neural network in precision griculture. *Journal of Engineering*, *21*(7), pp. 17-34. Doi:10.31026/j.eng.2015.07.02

Miao, Y., Tang, H., and Pan, G., 2018. A supervised multi-spike learning algorithm for spiking neural networks. *International Joint Conference on Neural Networks (IJCNN)*, pp. 1-7. Doi:10.1109/IJCNN.2018.8489175

Wu, D., Lin, X., and Du, P., 2019. An adaptive structure learning algorithm for multi-layer spiking neural networks. *15th International Conference on Computational Intelligence and Security (CIS)*, pp. 98-102. Doi:10.1109/CIS.2019.00029

Lee, C., Srinivasan, G., Panda, P., and Roy, K., 2018. Deep spiking convolutional neural network trained with unsupervised spike-timing-dependent plasticity. *IEEE Transactions on Cognitive and Developmental Systems*, 11(3), pp. 384-394. Doi:10.1109/TCDS.2018.2833071

Chen, Y., Rastogi, C., and Norris, W.R., 2021. A CNN based vision-proprioception fusion method for robust ugv terrain classification. *IEEE Robotics and Automation Letters,* 6(4), pp. 7965-7972. Doi:10.1109/LRA.2021.3101866

Vulpi, F., Milella, A., Marani, R., and Reina, G., 2021. Recurrent and convolutional neural networks for deep terrain classification by autonomous robots. *Journal of Terramechanics*, 96, pp. 119-131. Doi:10.1016/j.jterra.2020.12.002

Schilling, F., Chen, X., Folkesson, J., and Jensfelt, P., 2017. Geometric and visual terrain classification for autonomous mobile navigation. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* pp. 2678-2684. Doi:10.1109/IROS.2017.8206092

Karis, M.S., Razif, N.R.A., Ali, N.M., Rosli, M.A., Aras, M.S.M., and Ghazaly, M.M., 2016. Local Binary Pattern (LBP) with application to variant object detection: A survey and method. *IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA)*, pp. 221-226. Doi:10.1109/CSPA.2016.7515835

Singh, C., Walia, E., and Kaur, K.P., 2018. Color texture description with novel local binary patterns for effective image retrieval. *Pattern Recognition*, 76, pp. 50-68. Doi:10.1016/j.patcog.2017.10.021

Humeau-Heurtier, A., 2019. Texture feature extraction methods: A survey. *IEEE access,* 7, pp. 8975-9000. Doi:10.1109/ACCESS.2018.2890743

Kim, J., Kim, H., Huh, S., Lee, J., and Choi, K., 2018. Deep neural networks with weighted spikes. *Neurocomputing*, 311, pp. 373-386. Doi:10.1016/j.neucom.2018.05.087

Hu, Y., Tang, H., and Pan, G., 2021. Spiking deep residual networks. *IEEE Transactions on Neural Networks and Learning Systems*. Doi:10.1109/TNNLS.2021.3119238

Indiveri, G., Linares-Barranco, B., Hamilton, T.J., Schaik, A.V., Etienne-Cummings, R., Delbruck, T., Liu, S.C., Dudek, P., Häfliger, P., Renaud, S., and Schemmel, J., 2011. Neuromorphic silicon neuron circuits. *Frontiers in neuroscience*, 5, p.73. Doi:10.3389/fnins.2011.00073

Lin, X., Zhang, M., and Wang, X., 2021. Supervised learning algorithm for multilayer spiking neural networks with long-term memory spike response model. *Computational Intelligence and Neuroscience*, 2021. Doi:10.1155/2021/8592824

Al-Yassari, M.M.R., and Al-Jamali, N.A.S., 2023. Automatic Spike Neural Technique for Slicing Bandwidth Estimated Virtual Buffer-Size in Network Environment. *Journal of Engineering*, 29(6), pp. 87-97. Doi:10.31026/j.eng.2023.06.07

Yellakuor, B.E., Moses, A.A., Zhen, Q., Olaosebikan, O.E., and Qin, Z., 2020. A multi-spiking neural

network learning model for data classification. *IEEE Access*, 8, pp. 72360-72371. Doi:10.1109/ACCESS.2020.2985257

Oniz, Y., Kaynak, O., and Abiyev, R., 2013. Spiking neural networks for the control of a servo system. *IEEE International Conference on Mechatronics (ICM),* pp. 94-98. Doi:10.1109/ICMECH.2013.6518517

Thiruvarudchelvan, V., Crane, J.W., and Bossomaier, T., 2013. Analysis of SpikeProp convergence with alternative spike response functions. I*EEE Symposium on Foundations of Computational Intelligence (FOCI)*, pp. 98-105. Doi:10.1109/FOCI.2013.6602461

Nasser, F.K., and Behadili, S.F., 2022. Breast Cancer detection using decision tree and K-Nearest neighbour classifiers. I*raqi Journal of Science*, pp. 4987-5003. Doi:10.24996/ijs.2022.63.11.34

Abdulrezzak, S., and Sabir, F., 2023. An empirical investigation on Snort NIDS versus supervised machine learning classifiers. *Journal of Engineering,* 29(2), pp. 164-178. Doi:10.31026/j.eng.2023.02.11

Soud, N.S., and Al-Jamali, N.A.S., 2023. Intelligent congestion control of 5G traffic in SDN using dual-spike neural network. *Journal of Engineering*, 29(1), pp. 110-127. Doi:10.31026/j.eng.2023.01.07