

Enhancing Arabic Handwritten Character Recognition Using Multi-Reservoir Spiking Neural Network

Muhammad Raihaan Kamarudin^{1*}, Noorazlan Shah Zainudin², Zul Atfyi Fauzan¹, Sufry Muhammad³

¹Fakulti Teknologi Dan Kejuruteraan Elektronik dan Komputer, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

²Fakulti Kecerdasan Buatan dan Keselamatan Siber, Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia

³Fakulti Sains Komputer dan Teknologi Maklumat, Universiti Putra Malaysia, 43400, Selangor, Malaysia

*Corresponding author's

DOI: <https://dx.doi.org/10.47772/IJRISS.2025.909000220>

Received: 29 August 2025; Accepted: 04 September 2025; Published: 06 October 2025

ABSTRACT

Handwritten character recognition is an important area of artificial intelligence with applications in education, digital archiving, and cultural preservation. Arabic script recognition remains particularly challenging due to its cursive structure, positional variations of letters, and reliance on diacritical marks. This study introduces a multi-reservoir Spiking Neural Network (SNN) approach that mimics biological information processing to improve recognition performance. The proposed system integrates both original and augmented (Gaussian-blurred) representations of handwritten Arabic characters, enabling the network to capture diverse handwriting variations. Experiments conducted on a dataset of 16,800 samples demonstrate that the multi-reservoir model achieves higher accuracy than a single-reservoir baseline, particularly when applied to subsets of characters. Error analysis further reveals that most misclassifications occur among visually similar characters, highlighting the intrinsic complexity of Arabic script. These findings suggest that multi-reservoir SNNs provide a promising pathway for energy-efficient, culturally relevant AI applications. Beyond technical improvement, this work contributes to the digital preservation of Arabic language resources and supports broader access to information in multilingual societies.

Keywords: Spiking Neural Network, Arabic Handwriting Recognition, Artificial Intelligence, Multi-Reservoir Model, Digital Preservation

INTRODUCTION

Spiking Neural Networks (SNNs) are inspired by biological neurons that transmit information through discrete electrical pulses or spikes. Unlike conventional artificial neural networks (ANNs), SNNs encode information not only in neuron activations but also in the precise timing of spikes. This enables more biologically realistic processing and introduces temporal dynamics into computation. SNNs, often cited as the third generation of neural networks, have demonstrated improved energy efficiency and temporal data handling, particularly in image recognition and event-based vision tasks [1].

Character recognition broadly entails the automatic classification of textual content into meaningful categories. With the rapid growth of digital information, the efficacy of character recognition systems has become increasingly vital. Among the many scripts used worldwide, the Arabic alphabet holds particular significance, as it is employed by hundreds of millions of people across multiple nations. However, Arabic writing presents unique challenges for recognition due to its cursive nature, contextually varying character shapes, and reliance on diacritical marks [2]. These characteristics make segmentation and recognition of Arabic text more complex than that of Latin-based scripts, especially in the context of handwritten and historical documents [3].

To illustrate the complexity, Arabic OCR systems typically achieve reasonable accuracy for isolated characters but often fail when characters appear in initial, medial, or final positions within a word. Furthermore, the presence or absence of diacritical dots can dramatically alter the recognition outcome. These issues have limited the reliability of Arabic text recognition systems despite ongoing research efforts [2].

In this study, Spiking Neural Networks are explored as a framework for Arabic character recognition. The motivation for employing SNNs lies in their ability to leverage temporal encoding and unsupervised learning, making them particularly suitable for representing sequential features such as pen strokes. Mechanisms like Spike-Timing-Dependent Plasticity (STDP) enable SNNs to extract robust features directly from the temporal dynamics of input data [1], []. At the same time, the event-driven and binary firing nature of SNNs is highly compatible with neuromorphic hardware platforms, allowing for faster and more energy-efficient implementations compared to conventional neural networks [5]. Recent advances in spiking models, such as Spiking-YOLO, have shown that event-driven architectures can achieve competitive performance while reducing energy consumption by orders of magnitude [6].

Another factor contributing to the growing interest in SNNs is the development of simulation frameworks such as BindsNET, which is built on top of PyTorch. This framework facilitates rapid prototyping of SNN models on both CPU and GPU systems while providing support for biologically inspired learning rules such as Hebbian learning and STDP [7]. The availability of such tools lowers the barrier for deploying SNNs in practical machine learning applications, including reinforcement learning and character recognition [8].

The contributions of this work are as follows. First, we propose the application of spiking neural networks for handwritten Arabic character recognition, highlighting the ability of temporal dynamics to address the challenges posed by cursive structures and diacritical variability. Second, we implement and evaluate unsupervised learning mechanisms, specifically STDP-based training, to demonstrate that SNNs can learn meaningful representations from Arabic handwriting data. Third, we provide a comparative analysis of accuracy, computational efficiency, and scalability between SNN-based approaches and conventional recognition models. Finally, we present a modular implementation using the BindsNET framework to ensure reproducibility and enable future extensions. The remainder of this paper is organized as follows. Section II presents a review of related work in Arabic optical character recognition and spiking neural networks. Section III describes the dataset, preprocessing techniques, and network architecture. Section IV outlines the experimental setup and evaluation metrics, while Section V presents and discusses the results. Section VI concludes the study and highlights potential future directions.

RELATED WORKS

Spiking Neural Networks (SNNs) constitute the third generation of neural models, distinguished from earlier generations by their use of temporal spike-based coding and enhanced biological plausibility. This temporal dimension makes SNNs especially compelling for time-sensitive applications and hardware-friendly implementations. Consequently, researchers have explored various strategies to harness SNNs for character recognition and neuromorphic computing. One of the most influential early works is by Diehl and Cook [8], who applied unsupervised learning using Spike-Timing-Dependent Plasticity (STDP) to the MNIST digit classification task. Their model involved a two-layer architecture—input (28×28 neurons) and processing (excitatory and inhibitory)—achieving classification accuracies ranging from 82.9% to 95.0% by varying the number of excitatory neurons. This demonstrated the potential of SNNs in unsupervised representation learning. Extending this line of research, Hazan et al. [9] introduced a self-organizing SNN that merges features of self-organizing maps (SOMs) with spiking dynamics. The network achieves roughly 60% accuracy even after pruning 90% of synapses, illustrating how excitatory-inhibitory interactions and two-level inhibition mechanisms can support unsupervised filter map formation.

Diehl et al. [10] continued to push the boundaries by converting deep artificial neural networks into spiking equivalents. Utilizing a four-layer fully connected network trained with dropout and converted to an SNN, they achieved near state-of-the-art MNIST performance (98.68% test accuracy), showing that deep architecture conversion bridges the performance gap with traditional ANNs. Beyond digit recognition, Tavanaei et al. [11] provided a comprehensive review of training deep SNNs, comparing supervised and unsupervised learning

strategies and emphasizing computational efficiency—highlighting both progress and remaining challenges in training deep spiking networks. Bouvier et al. [12] surveyed hardware implementations of SNNs and their associated algorithmic adaptations, offering invaluable insight into real-world neuromorphic deployment and the existing gap between theory and practice. Eshraghian et al. [13] further contributed a tutorial-style perspective on integrating deep learning techniques into SNN training, including surrogate gradient methods to enable backpropagation-like optimization in spiking models. In the context of efficient event-based object detection, Kim et al. [14] introduced Spiking-YOLO, achieving comparable detection performance to Tiny YOLO on PASCAL VOC and MS COCO datasets, while offering up to 280× better energy efficiency on neuromorphic hardware.

Recent strides in closing the performance gap between SNNs and ANNs are exemplified in a Nature Communications study [15], where high-performance deep spiking neural networks were trained to match ANN accuracy—using fewer than 0.3 spikes per neuron, a leap in efficiency. On the application side of Arabic character recognition, deep learning methods have advanced rapidly. Amrouch and Rabi [16] used deep neural network features for Arabic handwriting recognition, delineating how deep representations improve recognition accuracy. In addition, Rabi et al. [17], who proposed a CNN-BLSTM (Bidirectional Long Short-Term Memory) model with CTC decoding for the KHATT dataset, showing enhanced performance on handwritten Arabic. Similarly, Elleuch et al. [18] examined the use of Deep Belief Networks for Arabic handwritten character and word recognition, demonstrating that regularization techniques such as dropout mitigate overfitting and boost performance. Other notable works include ensemble CNN approaches by De Haghighi et al. [19], achieving near 99.5% accuracy through model averaging, and residual network architectures by Finjan et al. [20] that reached accuracies up to 99.6% across several benchmark datasets, including MADBase.

More recently, Hussain et al. [21] proposed a hybrid Convolutional Spiking Neural Network (CSNN) model for Arabic handwritten digits. Leveraging both rate-based and STDP learning, they attained recognition rates of ~98.98% (CSNN) and ~91.16% (STDP-only SNN) on the ADBase dataset, highlighting the promise of spiking models in Arabic OCR tasks. These studies collectively reinforce the trajectory: from foundational SNN models using STDP to deep-learning guided spiking architectures, and from conventional CNNs for Arabic scripts to emerging spiking approaches tailored to language-specific challenges. However, while CNN and deep models dominate the Arabic OCR landscape, the integration of SNNs remains relatively limited—thus motivating our focus on SNNs for Arabic character recognition in this work.

DATA SET

The performance of any character recognition system is highly dependent on the quality and diversity of the dataset used. In this study, we employ a dataset of handwritten Arabic characters compiled from publicly available sources. The dataset was obtained through Kaggle, a widely used crowd-sourced platform for data science and machine learning challenges, which hosts a variety of curated datasets for academic and industrial use. The Arabic handwritten dataset used in this work consists of 16,800 character samples representing the 28 letters of the Arabic alphabet. These samples were contributed by 60 individuals between the ages of 19 and 40, with approximately 90% of the participants being right-handed. Each character appears in multiple handwritten variations, reflecting differences in stroke thickness, size, and diacritical marks. These variations capture the natural diversity of Arabic handwriting and thus present a realistic and challenging benchmark for recognition tasks. The example of the dataset is shown in Fig. 1.

For training and evaluation, the dataset was divided into three subsets: training, validation, and testing. The training set was used to optimize the spiking neural network parameters, while the validation set was employed to tune hyperparameters and prevent overfitting. Finally, the testing set served as an independent benchmark to assess generalization performance. Such partitioning is a standard practice in machine learning to ensure unbiased evaluation of model performance. Compared with Latin-based datasets such as MNIST, the Arabic dataset presents additional complexities due to its cursive writing style, positional variants of letters (initial, medial, final, and isolated forms), and the presence of diacritical dots that distinguish otherwise similar characters. These challenges make the dataset particularly suitable for assessing the robustness of Spiking Neural Networks, which can exploit temporal dynamics to learn discriminative features from noisy and highly variable input data.



Initial experiments using a single-reservoir Spiking Neural Network (SNN) architecture did not achieve the desired accuracy levels for Arabic character recognition. To address this limitation, a multi-reservoir structure was designed and implemented. This architecture enhances learning capacity by processing different representations of the input through parallel reservoirs, thereby improving feature diversity and recognition accuracy. In the proposed design, each image from the dataset undergoes a sequence of preprocessing and augmentation steps prior to encoding. First, the handwritten Arabic characters are converted to grayscale to normalize input variations and reduce computational overhead. The pixel intensity values are scaled within a fixed range (0–350), providing consistent input for spike encoding. To further enhance generalization, Gaussian blur augmentation is applied to the grayscale images. This step introduces controlled noise and mimics the natural variability observed in handwritten digits, similar to the blurriness of MNIST samples. By blurring the character strokes, the dataset contains more diverse pixel intensities rather than binary values, which helps the SNN learn richer feature representations.

Following preprocessing, the original grayscale image and its blurred counterpart are both encoded into spike trains using a Poisson encoder. The encoded spikes are then fed into two separate reservoirs: Reservoir 1 receives the original image, while Reservoir 2 receives the blurred version. This dual-reservoir strategy enables the network to learn complementary representations of the same character, thereby increasing robustness against noise, stroke variability, and diacritical mark differences. The encoded outputs are subsequently passed through the network for training and classification. The architecture is illustrated in Fig. 2, where the data flow proceeds from image preprocessing, augmentation, and encoding to dual-reservoir integration and classification. By combining features extracted from both reservoirs, the system achieves significantly higher accuracy compared with a single-reservoir baseline. Experimental results, discussed in Section V, confirm that the multi-reservoir structure enhances recognition rates, particularly when class sizes are reduced to subsets of the Arabic alphabet. An important aspect of this approach lies in its parameter sensitivity. Several hyperparameters directly influence performance, including:

- **Number of neurons** (n neurons): Increasing the reservoir size improves accuracy but at the cost of longer training time.
- **Number of classes** (n classes): Reducing the number of classes leads to higher recognition accuracy, highlighting the difficulty of full-scale 28-class classification.

- **Epochs** (n_{epoch}): Determines how many times the dataset is iteratively presented to the model.
- **Examples** ($training\ samples$): Larger training sets consistently yield better performance.
- **Spike duration** ($time$): Controls the length of the spike input window, directly affecting temporal learning.
- **Pixel intensity scaling** ($intensity$): Impacts the number of generated spikes, influencing learning stability.

Through systematic tuning of these parameters, we observed that increasing the number of neurons and training examples significantly improved accuracy, while moderate values for spike duration and intensity yielded the most stable performance. In summary, the multi-reservoir SNN design leverages data augmentation and parallel spike encoding to address the inherent challenges of Arabic handwritten character recognition. By integrating information from both original and blurred representations, the model demonstrates improved robustness and higher classification accuracy compared with conventional single-reservoir SNNs.

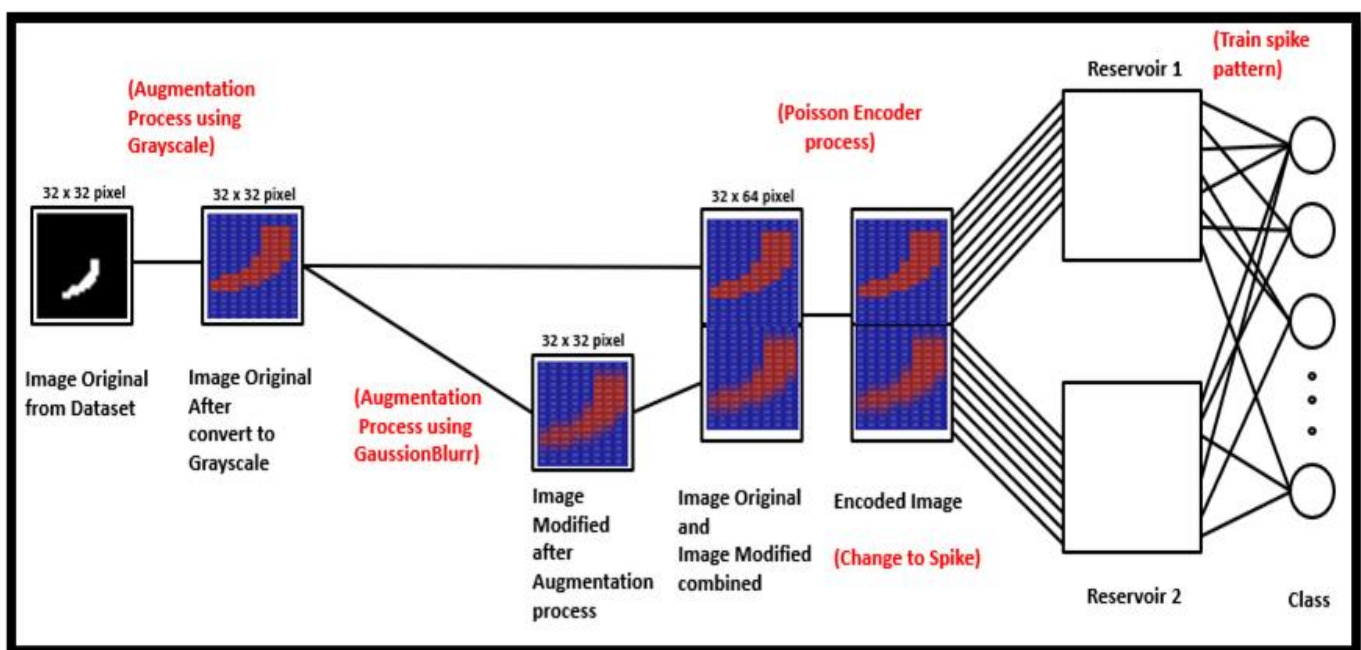


Fig. 2 Network Model Architecture and Overall Process

RESULT AND DISCUSSION

This section presents the experimental evaluation of the proposed multi-reservoir Spiking Neural Network (SNN) for Arabic handwritten character recognition. Results are analyzed across different parameter settings, augmentation strategies, and class configurations. Furthermore, classification errors are examined through confusion matrices to better understand the strengths and limitations of the model.

A. Parameter Sensitivity Analysis

To evaluate the effect of hyperparameters on recognition accuracy, experiments were conducted by varying the number of neurons, intensity, number of class and time. As shown in Table 1, the number of neurons plays an important role in recognition performance. Increasing neuron counts consistently improved accuracy; however, training time also grew substantially. A reservoir size of 500 neurons was found to provide an effective balance between accuracy and computational cost. Similarly, reducing the number of classes led to higher recognition accuracy, indicating that the proposed structure generalizes better when the classification task involves fewer character categories.

Table I: Prediction Accuracy on Different Parameter Setting Using All Class

Parameter	Value						
N neurons	600	600	600	500	700	1000	1000
N classes	28	28	28	28	28	28	28
N epoch	100	200	100	100	100	100	100
time	100	100	100	100	100	100	50
intensity	128	128	200	200	200	200	200
Accuracy (%)	57.5	55	62.6	60.7	63.3	64.4	59.8

B. Effect of Augmentation

Data augmentation was employed to increase robustness to handwriting variability. Three augmentation methods were tested: horizontal flip, vertical flip, and Gaussian blur. Here, blur produced the most consistent improvements compared with the other methods. Unlike flipping transformations, which alter the spatial orientation of characters and sometimes distort their natural structure, Gaussian blur preserved the fundamental shapes of the letters while adding variability in stroke intensity. This approach mimics natural variations in handwriting, thereby improving recognition performance without introducing unrealistic distortions. The effectiveness of Gaussian blur was further analyzed by varying kernel size and sigma parameters. Both were found to significantly affect classification accuracy, confirming that careful parameter tuning in the augmentation process is necessary to optimize system performance (refer Table 2).

Table II. Prediction Accuracy Effect Based on Augmentation Process

Kernel Size	(1,5)	(1,5)	(1,5)	(1,5)	(1,5)	(1,3)	(1,1)	(1,1)	(1,9)
Sigma	(0.5,2)	(1,2)	(0.5,1)	(0.5,0.5)	(1,1)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)	(0.5,0.5)
Accuracy (%)	61.0	60.4	62.1	63.3	60.8	61.2	60.8	60.9	59.3

C. Multi-Reservoir Classification Results

The proposed multi-reservoir structure was evaluated on the full dataset of 28 Arabic characters as well as on subsets with reduced class counts (10, 8, 6, 4, 2, and 1 classes). Results in Fig. 3 show that while recognition accuracy decreased with the full 28-class problem, it remained relatively high for smaller subsets. For instance, with only a single class, the system achieved 100% accuracy, whereas the 10-class configuration still maintained strong recognition rates. This demonstrates the ability of the multi-reservoir model to robustly classify Arabic characters, particularly when distinguishing among more distinct subsets of the alphabet.

Table III. Prediction Accuracy Based on Number of Class

Parameter	Value					
N neurons	1000	1000	1000	1000	1000	1000
N classes	10	8	6	4	2	1
N epoch	100	100	100	100	100	100

time	100	100	100	100	100	100
intensity	250	250	250	250	250	250
Accuracy (%)	86.4	90.3	90.0	95.4	99.2	100

D. Confusion Matrix Analysis

A detailed error analysis was performed using confusion matrices for both the 28-class and 10-class experiments. For the full 28-class configuration (Fig. 3), the highest recognition accuracy was observed for the character Alif (ا), with 109 correct classifications out of 120 samples. The lowest recognition rate occurred for Faa (ف), with only 49 correct classifications, often misclassified as visually similar characters such as Tow (ط) or Dod (ض). The total number of misclassifications across the 28-class dataset was 1310, indicating that although overall recognition is strong, there remains considerable confusion among characters with subtle stroke differences. For the 10-class experiment (Fig. 4), classification performance improved significantly. The character Alif (ا) again achieved the highest recognition accuracy, while the character Kof (ق) showed the most frequent misclassifications, particularly with Shin (ش) and Sod (ص). Notably, two characters achieved particularly low misclassification rates: Alif (4.12%) and Raa (ر) (1.67%). This confirms that the system performs best when the classes are visually distinct

LABEL	PREDICTED																													
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27		
	0	109	1	0	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	1	1	4	0	0	0	0		
	1	0	92	3	3	0	0	1	2	2	0	0	1	0	3	4	4	0	0	0	0	0	0	1	3	0	0	1		
	2	0	2	59	31	0	1	0	1	1	0	1	2	1	2	4	2	0	1	0	1	2	0	0	0	7	0	0	2	
	3	0	3	28	58	0	0	4	0	0	0	0	0	0	0	7	10	0	0	0	6	0	1	0	0	3	0	0	0	
	4	0	0	1	0	75	12	6	0	1	0	0	0	1	0	0	7	2	5	1	0	1	0	0	3	2	1	1	1	
	5	0	1	2	0	27	68	9	0	0	0	0	0	0	0	0	0	0	8	2	0	0	0	0	0	1	1	0	1	
	6	0	0	0	1	8	11	62	0	1	0	0	0	0	0	0	7	0	12	15	0	1	0	0	2	0	0	0	0	
	7	3	0	0	0	0	1	1	74	21	2	5	2	0	0	0	6	1	0	0	0	0	0	0	0	0	0	0	4	0
	8	0	0	0	0	0	0	1	13	74	0	3	0	0	0	0	20	0	2	0	0	0	1	0	0	0	0	6	0	
	9	1	0	0	0	0	0	0	2	1	97	15	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	2	0	
	10	0	1	0	0	0	1	0	2	7	18	70	0	1	0	0	6	0	0	1	1	0	1	3	0	0	0	8	0	
	11	0	3	2	0	2	0	1	0	0	0	74	6	16	6	1	0	0	0	4	1	1	0	0	0	1	0	2		
	12	0	0	1	1	1	0	1	0	0	0	1	76	0	18	11	0	0	0	3	3	1	0	1	1	0	0	1		
	13	0	2	2	1	0	1	0	0	0	0	0	10	3	72	14	10	1	0	0	2	0	0	0	1	0	1	0	0	
	14	0	0	1	0	1	0	0	0	0	0	0	3	8	11	63	14	0	1	0	8	0	0	0	0	5	0	1	4	
	15	0	0	0	0	0	0	1	0	1	0	0	0	0	1	1	86	22	0	0	1	0	0	0	0	0	5	2	0	
	16	0	0	0	0	2	0	2	0	0	1	1	0	1	0	2	48	50	0	1	2	1	0	0	1	0	3	5	0	
	17	0	0	1	0	8	6	12	0	0	0	0	0	0	0	0	5	1	72	10	0	0	0	0	2	1	1	0	1	
	18	1	0	0	0	2	2	18	0	0	0	2	0	0	0	0	10	2	16	65	0	0	0	1	0	0	1	0	0	
	19	0	1	11	0	0	0	2	0	1	0	0	0	0	4	14	17	1	0	0	49	10	1	0	0	2	1	3	3	
	20	0	0	4	1	0	0	4	0	0	0	0	0	7	3	5	17	0	0	1	14	53	2	1	0	0	0	2	6	
	21	0	1	4	2	0	1	0	0	0	0	0	0	0	1	8	7	3	0	1	1	2	75	1	0	5	4	3	1	
	22	1	2	0	0	0	0	1	0	1	0	0	0	2	0	1	2	1	0	1	2	0	7	94	0	3	0	1	1	
	23	3	0	0	0	2	0	0	0	0	1	1	1	0	3	0	5	1	0	0	1	0	0	0	101	1	0	0	0	
	24	0	3	5	2	1	0	0	3	3	1	0	1	1	0	10	8	1	2	0	5	4	5	3	1	60	0	0	1	
	25	0	0	1	0	2	0	0	0	1	0	0	0	2	1	0	25	6	1	0	1	0	0	0	1	0	69	9	1	
26	0	1	0	1	2	1	1	3	5	5	3	1	0	1	0	8	0	0	0	0	1	0	1	0	0	3	83	0		
27	0	6	0	0	5	2	2	1	0	0	0	2	5	2	5	6	2	0	0	3	2	1	3	0	1	1	1	70		

Fig 3. Confusion Table for 28 Class Prediction

LABEL	PREDICTED									
	0	1	2	3	4	5	6	7	8	9
0	115	1	1	1	0	0	0	2	0	0
1	0	109	0	0	2	8	0	1	0	0
2	0	0	105	0	1	5	3	4	0	2
3	0	0	0	118	0	1	0	1	0	0
4	0	0	2	0	102	6	1	0	6	3
5	0	4	4	0	2	107	0	0	1	2
6	2	0	1	2	8	4	90	2	4	7
7	1	0	3	0	3	0	3	107	3	0
8	0	2	3	0	21	9	3	0	80	2
9	1	1	2	5	5	8	5	0	1	92

Fig 4. Confusion Table for 10 Class Prediction

E. Sources of Misclassification

Further analysis revealed three major causes of errors.

1. Stroke similarity between characters: Characters such as Sod (ص) and Dod (ض) differ only by a diacritical dot, making them particularly confusable.
2. Structural overlap with other letters: Certain characters, such as Zow (ظ) and Tow (ط), share highly similar structures, leading to high misclassification rates.
3. Incomplete or noisy handwriting: In some samples, missing or extra strokes caused significant ambiguity in recognition.

These findings are consistent with previous studies on Arabic handwriting recognition, confirming that structural similarity and variability in diacritical marks pose major challenges (refer Fig. 5)

Similar characters	Master stroke
ب, ت, ث, د	ب
ح, ج, خ, د, ذ	ح
ر, ز	ر
س, ش, ط, ظ	س
ص, ض	ص
ع, غ	ع
ف, ق	ف
ك, ل	ل

Fig. 5 Arabic character stroke similarity

F. Discussion

Overall, the results demonstrate that the multi-reservoir SNN architecture improves recognition accuracy compared with a single-reservoir baseline. The combination of original and Gaussian-blurred inputs enhances the robustness of feature extraction, while parameter optimization and class reduction further improve performance. However, recognition of visually similar characters remains a challenge, indicating the need for additional feature extraction methods or hybrid learning strategies in future work.

CONCLUSION

In this work, a multi-reservoir Spiking Neural Network (SNN) was developed and evaluated for the task of Arabic handwritten character recognition. The proposed framework integrates both original and Gaussian-blurred representations of input images into parallel reservoirs, thereby enhancing feature diversity and improving classification performance. Experimental results demonstrate that the multi-reservoir structure achieves higher recognition accuracy compared with a single-reservoir baseline, particularly when the number of classes is reduced. The study also highlights the importance of parameter tuning. Increasing the number of neurons and training examples significantly improved recognition rates, although at the expense of computational complexity. Data augmentation, especially through Gaussian blur, proved effective in simulating natural handwriting variations and enhancing model robustness. Error analysis further revealed that most misclassifications occurred among characters with highly similar stroke structures or subtle diacritical differences, such as Sod vs. Dod or Zow vs. Tow.

Overall, the results confirm that SNNs, and in particular the multi-reservoir design, represent a promising direction for energy-efficient and biologically inspired approaches to Arabic character recognition. Nevertheless, challenges remain in distinguishing characters with minimal structural differences. Future work

will focus on integrating advanced encoding schemes, hybrid architectures combining SNNs with deep learning models, and deployment on neuromorphic hardware platforms to exploit the low-power advantages of spiking computation.

ACKNOWLEDGEMENTS

The author would like to thank Centre for Research and Innovation Management (CRIM), Universiti Teknikal Malaysia Melaka (UTeM) for its support in the present work.

REFERENCES

1. Yamazaki, K., Vo-Ho, V. K., Bulsara, D., & Le, N. (2022). Spiking neural networks and their applications: A review. *Brain sciences*, 12(7), 863.
2. Alghyaline, S. (2023). Arabic Optical Character Recognition: A Review. *Computer Modeling in Engineering & Sciences (CMES)*, 135(3).
3. Khorsheed, M. S. (2002). Off-line Arabic character recognition—a review. *Pattern analysis & applications*, 5(1), 31-45.
4. Maass, W. (1997). Networks of spiking neurons: the third generation of neural network models. *Neural networks*, 10(9), 1659-1671.
5. Furber, S. (2016). Large-scale neuromorphic computing systems. *Journal of neural engineering*, 13(5), 051001.
6. Kim, S., Park, S., Na, B., & Yoon, S. (2020, April). Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, pp. 11270-11277).
7. Hazan, H., Saunders, D. J., Khan, H., Patel, D., Sanghavi, D. T., Siegelmann, H. T., & Kozma, R. (2018). Bindsnet: A machine learning-oriented spiking neural networks library in python. *Frontiers in neuroinformatics*, 12, 89.
8. Diehl, P. U., & Cook, M. (2015). Unsupervised learning of digit recognition using spike-timing-dependent plasticity. *Frontiers in computational neuroscience*, 9, 99.
9. Hazan, H., Saunders, D., Sanghavi, D. T., Siegelmann, H., & Kozma, R. (2018, July). Unsupervised learning with self-organizing spiking neural networks. In *2018 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-6). IEEE.
10. Diehl, P. U., Neil, D., Binas, J., Cook, M., Liu, S. C., & Pfeiffer, M. (2015, July). Fast-classifying, high-accuracy spiking deep networks through weight and threshold balancing. In *2015 International joint conference on neural networks (IJCNN)* (pp. 1-8). iee.
11. Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T., & Maida, A. (2019). Deep learning in spiking neural networks. *Neural networks*, 111, 47-63.
12. Bouvier, M., Valentian, A., Mesquida, T., Rummens, F., Reyboz, M., Vianello, E., & Beigne, E. (2019). Spiking neural networks hardware implementations and challenges: A survey. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 15(2), 1-35.
13. Eshraghian, J. K., Ward, M., Neftci, E. O., Wang, X., Lenz, G., Dwivedi, G., ... & Lu, W. D. (2023). Training spiking neural networks using lessons from deep learning. *Proceedings of the IEEE*, 111(9), 1016-1054.
14. Kim, S., Park, S., Na, B., & Yoon, S. (2020, April). Spiking-yolo: spiking neural network for energy-efficient object detection. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 34, No. 07, pp. 11270-11277).
15. Stanojevic, A., Woźniak, S., Bellec, G., Cherubini, G., Pantazi, A., & Gerstner, W. (2024). High-performance deep spiking neural networks with 0.3 spikes per neuron. *Nature Communications*, 15(1), 6793.
16. Amrouch, M., & Rabi, M. (2017, April). Deep neural networks features for Arabic handwriting recognition. In *International Conference on Advanced Information Technology, Services and Systems* (pp. 138-149). Cham: Springer International Publishing.

17. Rabi, M., & Amrouche, M. (2024, April). Enhancing Arabic Handwritten Recognition System Based CNN-BLSTM Using Generative Adversarial Networks. In International Conference on Arabic Language Processing (pp. 140-153). Cham: Springer Nature Switzerland.
18. Elleuch, M., Tagougui, N., & Kherallah, M. (2015, March). Arabic handwritten characters recognition using deep belief neural networks. In 2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15) (pp. 1-5). IEEE.
19. Haghighi, F., & Omranpour, H. (2021). Stacking ensemble model of deep learning and its application to Persian/Arabic handwritten digits recognition. *Knowledge-Based Systems*, 220, 106940.
20. Finjan, R. H., Rasheed, A. S., Hashim, A. A., & Murtdha, M. (2021). Arabic handwritten digits recognition based on convolutional neural networks with resnet-34 model. *Indonesian Journal of Electrical Engineering and Computer Science*, 21(1), 174-178.
21. Hussain, N., Ali, M., Syed, S. A., Ghoniem, R. M., Ejaz, N., Alramli, O. I., ... & Ahmad, Z. (2024). Design and evaluation of Arabic handwritten digit recognition system using biologically plausible methods. *Arabian Journal for Science and Engineering*, 49(9), 12509-12523.
22. N. Hussain, H. Z. A. Jaffar, S. Aslam, and M. S. Farooq, "Design and evaluation of Arabic handwritten digit recognition system using biologically plausible methods," *Arabian Journal for Science and Engineering*, vol. 49, pp. 12153–12167, 2024. doi: 10.1007/s13369-023-08319-1.