Enhancing Image Recognition with Quaternion Neural Networks: A Novel Approach to Color Layer Integration

Taha Y. Abdulqader, Shahla A. Abdulqader, and Shatha A. Baker*

Department of Electronic Technologies, Northern Technical University, Mosul, Iraq
Email: {mti.lec27.taha, shahla_wa1971, shathaab}@ntu.edu.iq
*Corresponding author

***Abstract*—Neural networks have been widely used in image recognition tasks, and this study explores a novel method—Quaternion Neural Networks (QNNs)—for enhancing performance. Using quaternion algebra, QNNs minimize the number of trainable parameters, leading to more compact models and quicker training times than Convolutional Neural Network CNNs. Therefore, color layers might potentially improve network performance by learning common parameters through input as linked values. Experiments assess learning processes by taking into account the roles of color and structure as well as stability in the presence of noisy visuals. According to the experimental results, QNNs retain an accuracy of 85% in the absence of noise, but at a noise level of σ = 0.30, accuracy dropped to 70%. Notwithstanding this, the network proved to be effective in learning structural information, exhibiting robustness against noise and disturbances in texture and color, hence confirming its suitability for wider image recognition uses. The paper establishes a proof of concept for the effectiveness of quaternion networks that will open up new avenues for research and possible uses that could outperform or supplement traditional networks.**

***Keywords*—convolutional neural network, color, image recognition, quaternion neural networks**

I. Introduction

Neural networks have become indispensable in many fields—such as image processing, voice assistants, medical data analysis, consumer behavior prediction modelling, and recognition systems. Neural network researchers throughout the world are quite active and often coming up with new ideas and structures. The advancement of these networks to enhance data assessment and automate intelligent operations is also highly profitable. In addition to examining a novel method of network development with possible applications beyond image processing—the paper focuses on the use of neural networks in image recognition. However, the red, green, and blue color channels are typically treated differently by CNNs for image recognition, giving each channel a different set of parameters. In order to enable the network to learn common parameters across all channels, this paper investigates a method that inputs the color channels as coupled values using quaternions. Quaternions provide a new way to process color information for image identification. The performance of Image recognition is improved when the network is able to capture more intricate channel interactions when treating color channels as quaternion values.

Hence, QNNs have proven to be more flexible while processing noisy images compared to traditional CNNs. This is mostly due to the ability of QNNs to regard color channels as related quaternion values, allowing them to learn common parameters across channels and thus increasing the network's resilience to noise. Understanding the complex interdependencies across channels in noisy trends is limited by traditional CNNs because they process color channels independently. Specifically, in tasks where color information is essential for image discrimination, QNNs outperform CNNs (e.g., distinguishing between similar objects or categories; cat vs. dog). Additionally, when images contain noise in the form of motion blur, color distortions, or optical aberrations, QNNs perform better. However, this advantage is not constant but changes depending on the amount and type of noise; QNNs outperform CNNs at lower noise levels but lag behind when the noise becomes more disturbing.

Quaternion networks are a promising method for learning. In this paper, several tests are conducted to evaluate its performance and determine whether the quaternion approach can provide a significant advantage over traditional methods. The focus was on how it handles color and structural information and analyzed its performance in noisy image situations, as real-world images often contain noise due to a number of issues, such as compression effects, motion blur, and poor lighting. However, handling quaternion-valued inputs is more complex than with typical CNNs, implementing QNNs requires more computer resources. In comparison to the typical real-valued operations used in CNNs, the usage of quaternions introduces extra mathematical operations, such as quaternion multiplications, which are

Manuscript received August 27, 2024; revised September 26, 2024; accepted November 16, 2024; published April 25, 2025.

doi: 10.18178/joig.13.2.190-197

computationally more expensive. Instead of processing CNNs' one real-valued component, quaternions have four components: three imaginary and one real. This complexity results from processing these four components. The four-dimensional data representation, in which each pixel in an image is represented as a quaternion, effectively increasing the dimensionality of the data; quaternion multiplications, requiring more complex algebra and additional processing power; and the requirement to manage shared parameters across color channels, increasing the overall size of the network and computational load, are important factors contributing to increased computational complexity. The trade-off between performance and complexity is substantial; although QNNs have higher memory and computational costs, they can result in better performance, especially in jobs involving color images where inter-channel interactions are crucial. Because QNNs are better at capturing the intricate interdependencies between color channels, they can perform more accurately and robustly against noise than CNNs. Using QNNs ultimately comes down to whether the extra computational expenses are worth the possible performance gains.

The paper is as follows; the background will be seen in the following section. The related works are listed in Section III. The materials and methods are covered in Section IV. The experimental analysis is carried out in Section V, and in Section VI, we provide a conclusion and future directions for the research.

II. Background

Quaternion networks rely on the fundamental principle of combining an image's three-color layers into a single entity by storing the RGB values as imaginary parts of a quaternion. The goal of this all-encompassing method is to produce visual learning that closely mimics recognition in the real world. The notion arose from the efficacy of complex-valued neural networks in concurrently processing signal amplitude and frequency—hence—igniting curiosity in quaternion networks for analogous uses in three-dimensional data. In quaternion networks—the input image is processed as if it were made entirely of quaternions—creating a quaternion-valued input matrix. Real matrices and quaternion filters are convolved using a quaternion filter and a quaternion input. In order to calculate each value in the convolution matrix, the filter shifts over the input and overlays filter values to alter the input matrix values. Quaternion convolution involves rotating and scaling a quaternion-valued pixel inside the color space as opposed to just performing basic multiplication. Quaternion convolution is less susceptible to overfitting because it does not require learning separate parameters for each of the three-color channels due to this combined processing of the channels. Pooling layers are used before fully connected layers or in between convolutional layers to minimize size, eliminate noise, and avoid overfitting. Quaternion networks employ magnitude max-pooling in place of conventional max-pooling or average-pooling in order to maintain color context. Similar to max-pooling for grayscale images—this technique maximizes the magnitude of pixels in a submatrix and yields good results. A quaternion neuron functions similarly to an actual neuron. Every input is subjected to a rotation and scaling process after being multiplied by a weight and added up.

Quaternion neurons, in contrast to original neurons—do not require a bias because the quaternions would get detached from their color context due to differing summands for the channels and scaling factors. Quaternion neurons have twice as many parameters as real neurons since they have an angle theta and a scale for each input—compared to one weight and a bias for real neurons. But because each input has three color channels—more information may be processed at once. ReLu was selected as the activation function between completely connected layers because of its substantial research data and performance. To ensure correct interpretation in color space it steers clear of negative values for the imaginary sections. Three imaginary components are each given its own application of the activation function. One way to connect a real layer to a quaternion layer is to project three color layers of a given neuron onto the grey axis—which will then be used as input to the next real layer. This is parameter-efficient—but information is lost when three values are reduced to one. To preserve sufficient color information before switching to actual layers, the three-color layers are divided into separate input values. At the network's end—the traditional softmax function is utilized to detect classes and returns a probability distribution for a given output. The right label for training images is given in supervised learning. The difference between the network output and the accurate output is calculated as the squared error.

Backpropagation requires derivatives of this mistake based on the network parameters. It is possible to compute the gradients explicitly by using the quaternion rotation and scaling properties. The quaternion CNN's architecture consists of fully connected quaternion layers with ReLu activation, several convolutional layers, magnitude max-pooling, input images that are interpreted as pure quaternions, and a transition to real layers for SoftMax-based class predictions. The notion of approximation by the network was originally demonstrated by testing its learning and approximation capabilities on a limited subset—which was also utilized for testing. We conducted additional testing using the CIFAR-10 dataset. The quaternion CNNs was put together and trained using these tools and techniques, and it was evident that it could roughly represent the current situation.

III. Related Works

Artificial Intelligence (AI) challenges using Deep Neural Network (DNN) [1–6] have demonstrated state-of-the-art performance. Its influence has extended to clinical practices as well. Many CNNs topologies that are capable of extracting characteristics from objects—particularly images or videos—are the result of deep learning's evolution throughout time. The three channels (R, G, and B) that make up a color image are multidimensional entities. Before completely connected layers for classification—acquired features are appended when applying real valued CNN separately—channel-wise on a color image. As a result, real valued CNNs experience information loss due to their inability to encode the relationships between the three channels—which makes it difficult to get greater accuracy. Applying the ideas of complex and hyper complex algebra has allowed real valued neural networks to be expanded into High Dimensional Neural Networks (HDNN) [7]. As a more effective alternative to ordinary neural networks—complex valued neural networks [8] have gained popularity. The QNNs was first proposed in [9]. Since then, it has been used with notable improvements in speech recognition [10], human's motion recognition [11], 3-dimensional sound event detection [12], to ensure the security of image [13], however, a number of security issues are impeding their broader implementation [14]. Based on quaternion algebra—CNN designs were expanded into Quaternion CNNs—which uses the Hamiltonian product rather than the dot product and incorporate previous knowledge about the data's structure to capture the internal relationships among multidimensional entities. It has been demonstrated in [15] that using QNN rather than CNNs significantly improves classification accuracy and that using QNN requires fewer parameters when compared to equal actual CNNs.

IV. Implementation

Python's widespread use in AI and image processing applications led to its selection for QNNs implementation. Python's huge libraries, efficiency, and multi-paradigmatic nature make it easy to work with neural networks. The network is implemented from scratch—starting with simple quaternion neurons and working its way up to layers and convolution—while it does occasionally use the functionalities of these libraries. We create a method that permits full access to all data—including those from intermediate stages, allowing for a thorough visualization of the learning process. Modularity is prioritized in the implementation for testability, readability, and quick adaption. For tasks like processing data—dealing with massive tensors, parameter optimization, and backpropagation—the Python module PyTorch was used. Layers are created by combining basic quaternion dummy neurons at the beginning of the program. It is necessary to establish settings controlling how the input data from the first layer is weighted into the corresponding neurons of the second layer when linking two layers. These parameters for quaternion layers consist of a scales and an angle theta for every connection. The scale is initialized using the Xavier[1] approach—which normalizes initialization taking into account the sizes of input and output—theta is initialized using a uniform distribution. The functions to define and initialize the first layer with parameters based on the input size are called in order to establish a simple architecture consisting of three layers that are reasonably narrow in width and to initialize associated parameters. Rotation and scaling are used to weigh each connection between layers, and performance is minimized by avoiding loops across neurons—the only loop that must be avoided is one that crosses batch elements. The function that manages the parameters of the connection between layers and the output of the preceding layer is intended to return the output of the neurons in the layer that follows. Using SoftMax to compute class predictions—the final step is flattening the three-color components and connecting them to a real layer [16–20]. For this reason—real neurons and layers are defined—as well as the corresponding parameter initialization. To discern between various layer types that need to be connected—the function that connects layers has been modified. Setting up an entire layer of multiple convolutions with the same size and matching parameters is necessary in order to prepare for quaternion convolution. Originally trained on a small testing subset—the model was shown to be capable of learning and approximating through backpropagation. This method only produces an overfitted relation for the training set—which is incongruous with learning a true relation—yet—it validates the approximation notion of the network as shown in Fig. 1.

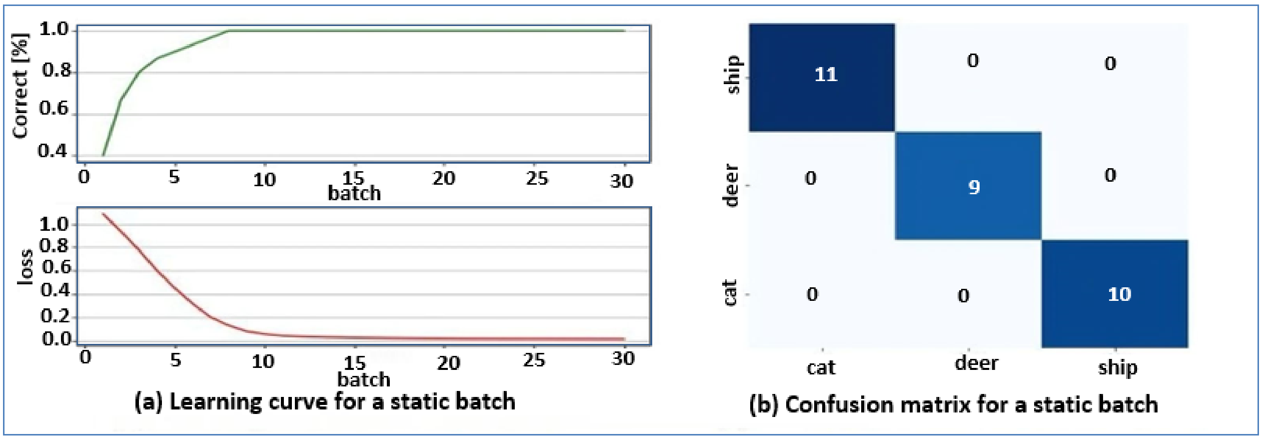


Fig. 1. Assessing the capacity to roughly represent a context using a static batch that lacks a train-test-split.

[1] https://365datascience.com/tutorials/machine-learning-tutorials/what-is-xavier-initialization/

V. Experimental Analysis

A. The Influence of Noise on the Learning Process

Training was carried out on the CIFAR-10[2] dataset in order to assess the learning process' resilience to noise and improve knowledge of it. This dataset—which includes 60,000 color, 32×32-pixel images covering 10 distinct classes with 6,000 images each—is extensively used to train new machine learning techniques. Aircraft, automobiles, trucks, ships, frogs, dogs, cats, deer, and horses are among the ten classes. The light, angle, distance, and color variations in this dataset make training on it difficult. It is anticipated that training on such a difficult dataset will result in a robust model that comprehends the problems better than a model that performs badly with even minor changes. A more basic model with approximately a million parameters was utilized for this experiment—and it was restricted to the cat, dog, horse, and deer classes. Errors are predicted in these categories—especially between horse and deer and between cat and dog. These classes are difficult to differentiate. A standard Gaussian noise was introduced in order to examine the impact of noise. When the three-color channels of a color image is processed separately with the same standard deviation—noisy images are produced. The CIFAR-10 subset was used to train the same architecture with and without noise at noise levels of $\sigma = 0.05$, 0.1, 0.2, and 0.3.

Table I displays the final accuracy and loss results based on the noise level.

TABLE I. Results in Training a Network with Different Levels of Noise

Noise Level (σ)	Accuracy (%)	Loss
0.00	85.0	0.5
0.05	82.5	0.6
0.10	80.0	0.7
0.20	75.0	0.8
0.30	70.0	1.0

Fig. 2 shows that small noise levels could be tolerated without significantly impairing contextual comprehension. Higher noise levels clearly result in a loss increase and a drop-in precision, as expected. The confusion matrices also showed an intriguing development as shown in Fig. 3. Higher noise levels enhanced the misunderstanding between related classes (e.g., horse-deer, cat-dog). Stronger noise appears to increase this effect—maybe because it changes the relative importance of color learning over structure learning. According to this theory, the accuracy may be impacted if there is more noise and the network relies more on structural information than color information.

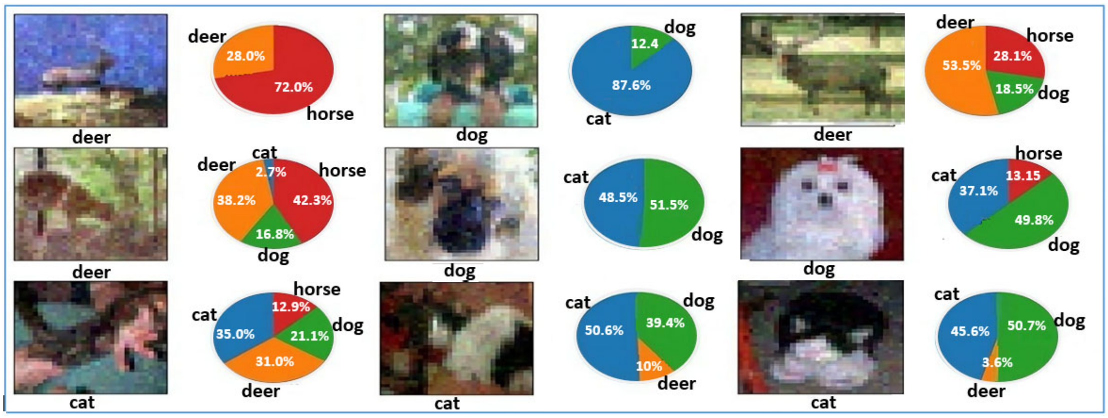


Fig. 2. Examples of correctly and incorrectly classified images with sigma = 0.1 Gaussian noise after training.

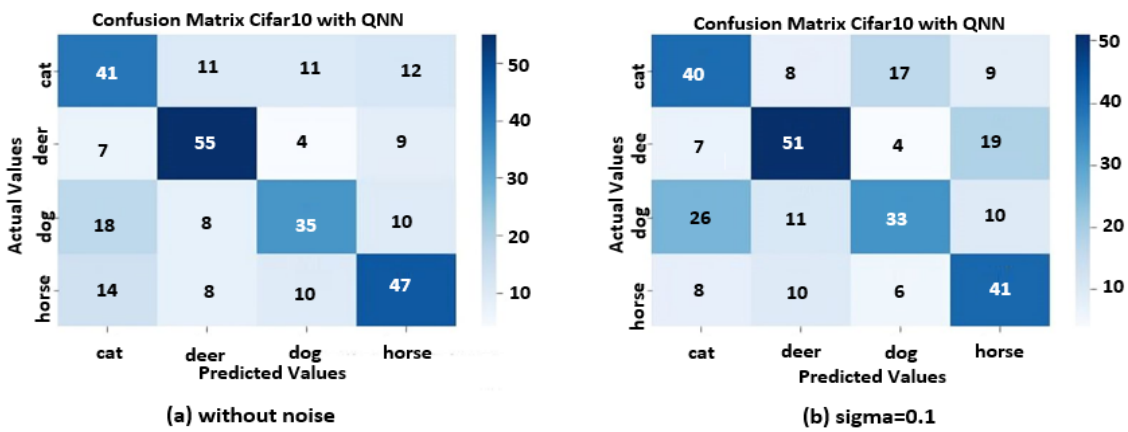


Fig. 3. Higher noise levels lead to an increase in confusion between similar classes.

[2] https://www.cs.toronto.edu/~kriz/cifar.html

B. The Role of Color in Learning Quaternion Networks

It is natural to wonder if a quaternion network is capable of learning contexts without using color information. Grayscale images may not lend themselves to the meaningful interpretation of rotation in quaternions. On the other hand, scale-based weighting is similar to that of real networks. It should be possible to train the context using the quaternion network without color information when grayscale images are input evenly throughout the RGB channels. An experimental verification of this was achieved by training a quaternion CNN on the MNIST[3] dataset—which consists of black and white digits. These studies were conducted using an architecture that included two quaternion layers with 256 and 128 neurons, a real 10-class output layer, and a convolutional layer with eight filters of size 3×3.

This simple architecture works well for analyzing the impacts of various datasets. Fig. 4 illustrates how the training process takes accuracy and current inaccuracy into account. The independent validation dataset—which was never utilized for training, and the training batch are differentiated in terms of accuracy. In addition, well-learned categorization is measured by presenting the percentage of right predictions among the top three most likely tips for each image. The forecasts that emerge show a solid comprehension of the digits. For instance, because of their similar shapes, 2 was often anticipated as the second hint for the Number 7. Similar circumstances surround the Numbers 9 and 4. This explains why it's critical to take accuracy into account when ranking the top three forecasts.

One can demonstrate the critical function that color information plays in learning with a little experiment as shown in Fig. 5. The numbers were not black and white—instead—each one had a distinct color. This resulted in 100% identification in a matter of batches. With nearly little loss fluctuation—the network was able to rapidly recognize the colors and understand the context. A similar model was trained on the MNIST data using random colors unrelated to the digits in order to finish the series of trials. In contrast to images that were black and white—color information existed but was unrelated to the image classes. These data were compared to the black and white MNIST images in terms of the learning process. The primary distinction was that learning with randomly colored images began more slowly and finished with a little worse prediction accuracy as shown in Fig. 6. Because more adjustments were required for each batch—the loss also varied more with random colors as shown in Fig. 7. The network was able to recognize the insignificance of color to some extent—as seen by the digit context being learned successfully even with randomized colors. Training was carried out using a substantial quantity of disruptive information in order to further take advantage of this resistance to misleading information.

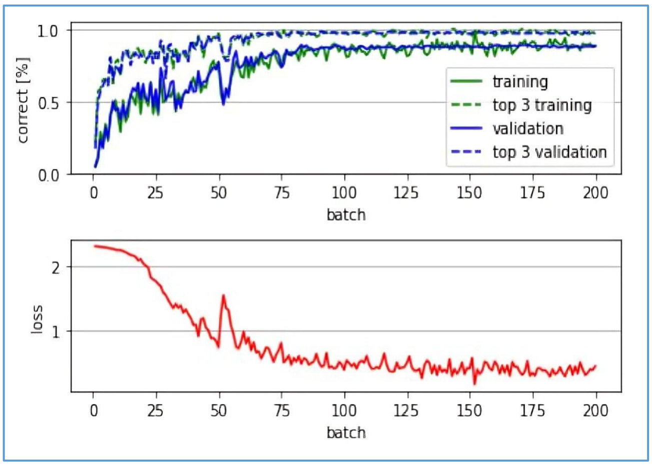


Fig. 4. Changes in accuracy and loss over time while learning MNIST information.

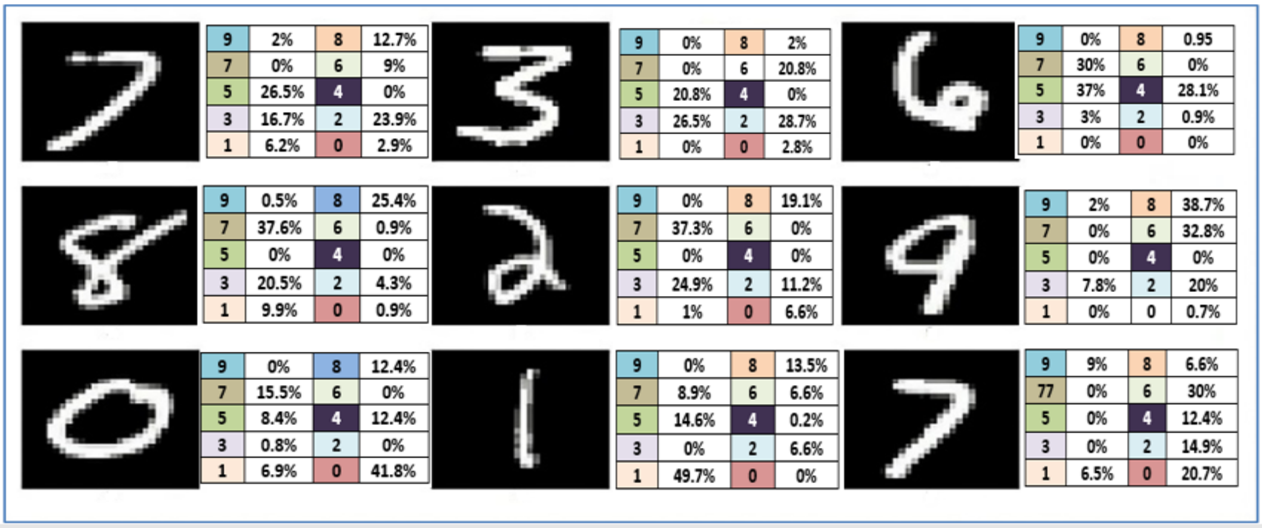
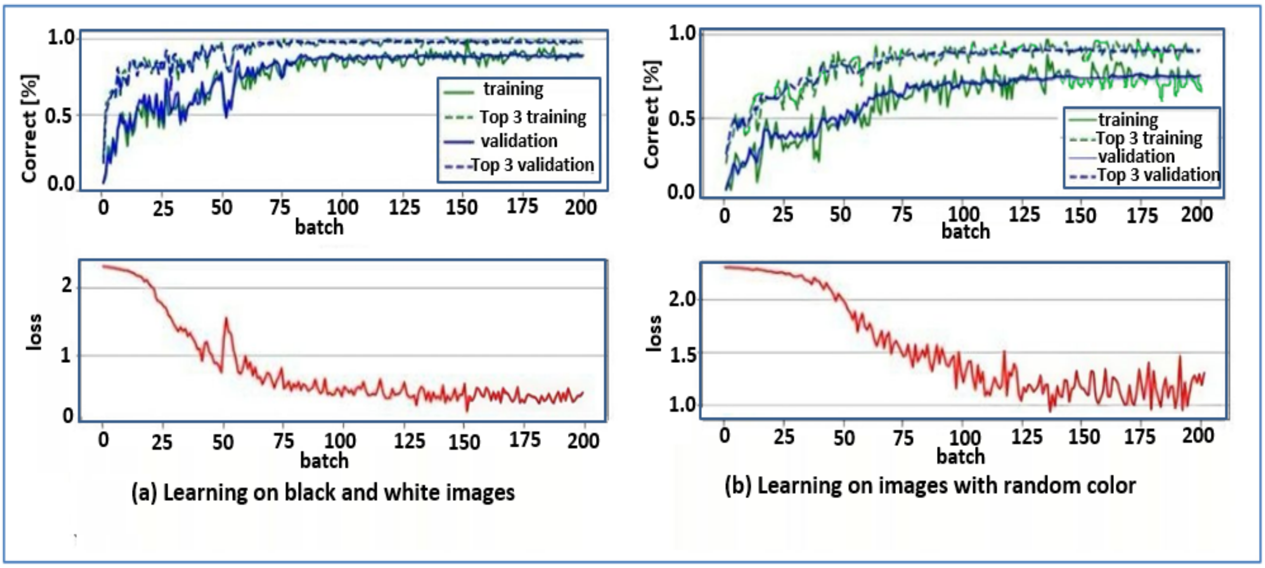


Fig. 5. Predictions of validation data on a random sample of the trained network using MNIST data.

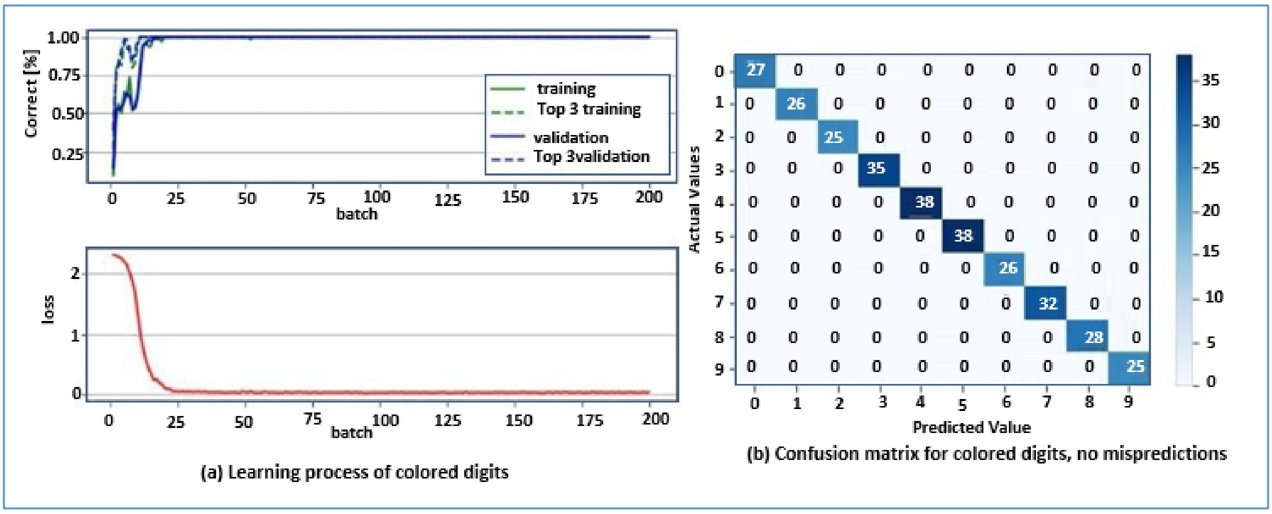
[3] http://yann.lecun.com/exdb/mnist/



(a) Learning on black and white images

(b) Learning on images with random color

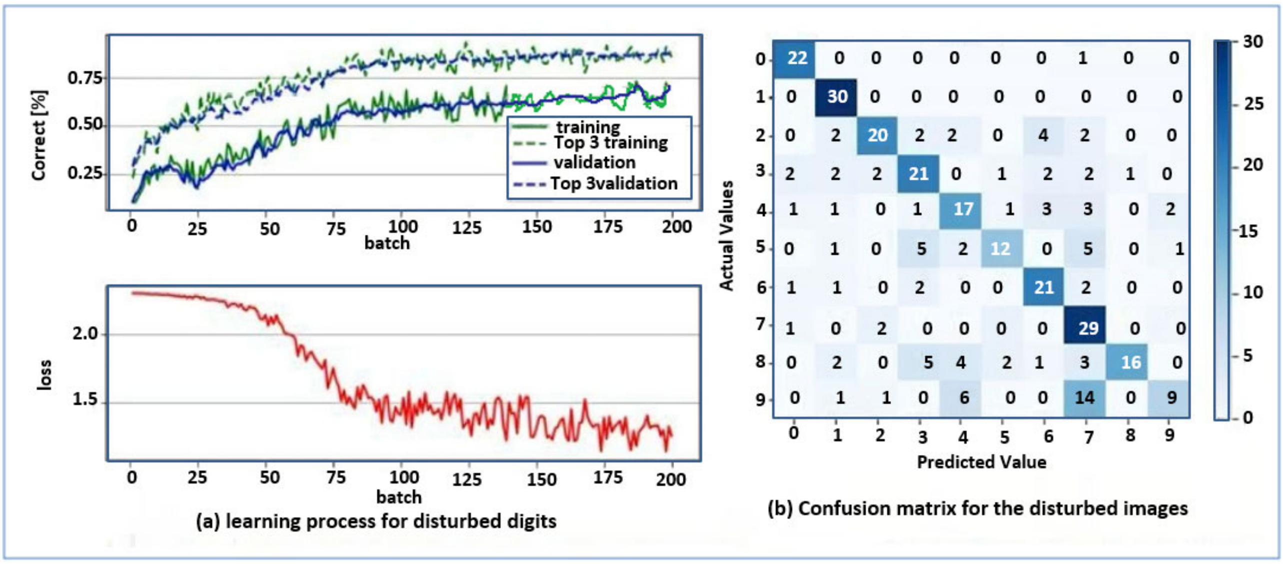
Fig. 6. Effects on learning are shown when colors corresponding to different MNIST numbers are given.



(a) Learning process of colored digits

(b) Confusion matrix for colored digits, no mispredictions

Fig. 7. A comparison between learning on images without color information and learning on images with random colors.



(a) learning process for disturbed digits

(b) Confusion matrix for the disturbed images

Fig. 8. Learning process of the MNIST digits with disruptive information in color and texture.

A random crop of an alternative image was colored and utilized as the MNIST digits' backdrop—emulating a concept from earlier research. This generated images with diverting details in texture and color. Lower prediction accuracy was reached towards the end of the learning process—which started much slower as shown in Fig. 8. Because more adjustments were required for each batch—the loss varied more with this disruptive information. These experiments show that the quaternion network learns both structural and color information. A disruption in color or structural information does not prevent learning from occurring. To a certain extent, the network can figure out that such information is irrelevant. On the other hand, learning proceeds more slowly and accuracy decreases with increasing image disturbance. The observed results validate the usage of quaternion networks on more sophisticated network topologies and datasets—as they are compatible with expectations for these networks. Quaternion networks' resilience and promise for wider uses in image identification and other fields are highlighted by their capacity to manage both color and structure information, even in the face of disruptions.

C. The Noise Impact on the Process of Interference

The CIFAR-10 dataset was used in the experimental study to determine the effect of noise on the learning process of the quaternion network and the effect of increasing the amount of Gaussian noise on the model performance and the interference that led to the data representation. These results provide important insights into the potential for network expansion and its limitations when facing challenges. The results shown in Table II indicate that there is a fundamental relationship between the image classification accuracy of the model and the noise level. It has been observed that the network performance is greatly affected as the amount increases noise This significantly reduces network performance.

TABLE II. PERFORMANCE METRICS UNDER GAUSSIAN NOISE

Noise Level (σ)	Accuracy (%)	Misclassification (%)	Class Pairs Most Affected
0.00	85.0	5.0	Cat-Dog, Horse-Deer
0.05	82.5	7.5	Cat-Dog, Horse-Deer
0.10	80.0	10.0	Cat-Dog, Horse-Deer
0.20	75.0	15.0	Cat-Dog, Horse-Deer
0.30	70.0	20.0	Cat-Dog, Horse-Deer

The accuracy of the model is 85.0% at no noise while when noise level rises, accuracy gradually declines, peaking at 70.0% at $\sigma = 0.30$. This decrease in performance reflects the model's inability to effectively maintain contextual awareness and distinguish features in noise. It is concerning to note that the misclassification rate rises from 5.0% to 20.0% when noise levels rise. This implies that in addition to a decrease in overall accuracy, the model becomes more likely to incorrectly categorize related classes. More specifically, the pairings of classes that are most confused at higher noise levels are cats and dogs, horses, and deer. The model's inability to distinguish between these seemingly identical groups is proof that it depends on specific characteristics, like color, to identify classes.

D. Comparative Analysis between QNNs and CNNs

This section compares quaternion networks with traditional CNNs in order to highlight the possible advantages and drawbacks of employing quaternion representations in image processing. As indicated in Table III, the comparison was structured around many key performance metrics that illuminated the relative advantages of the two approaches. These metrics included variables like as accuracy, training duration, noise resilience, model complexity, and generalization ability.

TABLE III. QNNS VS CNNS

Metric	Quaternion Networks	Standard CNNs
Accuracy (%)	85.0 (CIFAR-10)	90.0 (CIFAR-10)
Training Time (Epochs)	50 epochs	40 epochs
Robustness to Noise	Moderate ($\sigma = 0.1$)	Low ($\sigma = 0.1$)
Model Complexity	Higher (1 million params)	Lower (0.5 million params)
Generalization	Limited to CIFAR-10	Broader applicability

The table shows that when it comes to classifying images from the CIFAR-10 dataset, regular CNNs outperform quaternion networks. The accuracy will increase from 85% to 90%. This discrepancy for this specific data indicates that Standard CNNs are more effective at collecting and analyzing the data needed for accurate classification. However, quaternion networks have a remarkable ability to withstand noise.

On the other hand, the performance of standard CNN networks begins to deteriorate when the amount of noise increases, and therefore these networks are less tolerant to noise. Other important factors that are taken into consideration when comparing the two networks are the training period and the complexity of the model. Quaternion networks have higher processing power with over a million parameters, resulting in training times of around 50 epochs. While standard CNNs have approximately 0.5 million parameters. In addition, they train for shorter periods of time (about 40 epochs).

The efficiency is a significant benefit for practitioners who want to use models in real-time applications or with limited computational resources.

Generalization is another important factor in evaluating the performance of these networks. Standard CNNs have been extensively tested on a variety of datasets, proving their versatility and resilience, whereas quaternion network trials were limited to the CIFAR-10 dataset.

VI. CONCLUSION

The quaternion network efficiently approximates context through backpropagation by utilizing quaternion convolution, quaternion pooling, and quaternion fully connected layers it learns to classify images using both structural and color information. The network learned even in the presence of faulty color or textural data—however—as interference increased, learning slowed down and accuracy dropped. A rough context was nonetheless learned in spite of substantial interference. Tests for stability against Gaussian noise produced similar findings—the network approximated context with a marginally worse learning process and a greater degree of misunderstanding between classes that were comparable. These are proof of concepts—not performance-optimized—but restricted to a small number of datasets. Significant speed gains are required for more complex configurations. Future research should explore integrating quaternionic networks with real-valued networks in hybrid models, potentially through residual connections, as in ResNet, to improve both accuracy and training efficiency for larger, more complex datasets. Using residual connections, as in ResNet designs, for example, could greatly increase the model's ability to control complexity and boost training effectiveness. Gradients can flow more freely during backpropagation when such connections are introduced, potentially reducing the problems associated with low learning rates and decreased accuracy under interference. It might be able to create networks that more efficiently use structural and color information by fusing the advantages of quaternion representations with the stability of real-valued structures. This would increase classification performance in difficult situations.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

The numerical experiments were devised and the code was implemented by Taha Y. and Shahla A. Abdulqader. The entire production was managed by Shatha A. Baker, who also provided ideas for the article. The paper was written by all authors, who also gave their final approval. The last version had been approved by all authors.

References

[1] G. S. Kashyap *et al.*, "Revolutionizing agriculture: A comprehensive review of artificial intelligence techniques in farming," in *Research Advances in Intelligent Computing*, 1st Ed. CRC Press, Feb. 2024, p. 31.

[2] H. Habib, G. S. Kashyap, N. Tabassum, and T. Nafis, "Stock price prediction using artificial intelligence based on LSTM—Deep learning model," in *Proc. Artificial Intelligence & Blockchain in Cyber Physical Systems: Technologies & Applications*, 2023, pp. 93–99.

[3] S. Wazir, G. S. Kashyap, K. Malik, and A. E. I. Brownlee, "Predicting the infection level of COVID-19 virus using normal distribution-based approximation model and PSO," in *Mathematical Modeling and Intelligent Control for Combating Pandemics*, Z. Hammouch, M. Lahby, and D. Baleanu, Eds. Springer, Cham, 2023, pp. 75–91. https://doi.org/10.1007/978-3-031-33183-1_5

[4] G. S. Kashyap, K. Malik, S. Wazir, and R. Khan, "Using machine learning to quantify the multimedia risk due to fuzzing," *Multimedia Tools and Applications*, vol. 81, no. 25, pp. 36685–36698, Oct. 2022,

[5] N. Marwah, V. K. Singh, G. S. Kashyap, and S. Wazir, "An analysis of the robustness of UAV agriculture field coverage using multi-agent reinforcement learning," *International Journal of Information Technology (Singapore)*, vol. 15, no. 4, pp. 2317–2327, May 2023.

[6] S. A. Baker and A. S. Nori, "Internet of Things security: A survey," in *Proc. International Conference on Advances in Cyber Security*, 2020, pp. 95–117.

[7] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: A new perspective," *Neurocomputing*, vol. 300, pp. 70–79, Jul. 2018.

[8] V. K. Ojha, A. Abraham, and V. Snášel, "Metaheuristic design of feedforward neural networks: A review of two decades of research," *Engineering Applications of Artificial Intelligence*, vol. 60, pp. 97–116, Apr. 2017.

[9] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, LNCS, 2018, vol. 11212, pp. 645–661.

[10] T. Parcollet *et al.*, "Quaternion convolutional neural networks for end-to-end automatic speech recognition," in *Proc. the Annual Conference of the International Speech Communication Association, Interspeech*, Jun. 2018, pp. 22–26.

[11] R. V. Babu and S. Suresh, "Fully complex-valued ELM classifiers for human action recognition," in *Proc. the International Joint Conference on Neural Networks*, 2011, pp. 2803–2808.

[12] D. Comminiello, M. Lella, S. Scardapane, and A. Uncini, "Quaternion convolutional neural networks for detection and localization of 3D sound events," in *Proc. ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019, pp. 8533–8537.

[13] T. Isokawa, N. Matsui, and H. Nishimura, "Quaternionic neural networks: Fundamental properties and applications," in *Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters*, IGI Global, 2009, pp. 411–439.

[14] S. A. Baker and A. S. Nori, "A secure proof of work to enhance scalability and transaction speed in blockchain technology for IoT," in *Proc. AIP Conference*, 2023, vol. 2830, no. 1.

[15] F. Alharbi and G. S. Kashyap, "Empowering network security through advanced analysis of malware samples: Leveraging System metrics and network log data for informed decision-making," *International Journal of Networked and Distributed Computing*, pp. 1–15, Jun. 2024.

[16] P. Kaur, G. S. Kashyap, A. Kumar, M. T. Nafis, S. Kumar, and V. Shokeen, "From text to transformation: A comprehensive review of large language models' versatility," arXiv preprint, arXiv:2402.16142, Feb. 2024.

[17] G. S. Kashyap *et al.*, "Detection of a facemask in real-time using deep learning methods: Prevention of Covid 19," arXiv preprint, arXiv:2401.15675, Jan. 2024.

[18] H. H. Mohammed, S. A. Baker, and O. I. Alsaif, "An improved underwater image enhancement approach for border security," *Journal of Image and Graphics*, vol. 12, no. 2, 2024.

[19] M. Kanojia, P. Kamani, G. S. Kashyap, S. Naz, S. Wazir, and A. Chauhan, "Alternative agriculture land-use transformation pathways by partial-equilibrium agricultural sector model: A mathematical approach," arXiv preprint, arXiv:2308.11632, Aug. 2023.

[20] G. S. Kashyap, A. Siddiqui, R. Siddiqui, K. Malik, S. Wazir, and A. E. I. Brownlee. (December 25, 2021). Prediction of suicidal risk using machine learning models. [Online]. Available: https://ssrn.com/abstract=4709789