International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 14 (2022) pp. 262-269

Received: 2 March 2021; Accepted: 8 February, 2022; Published: 28 April, 2022

Deep Quaternion Residual Learning for Breast Cancer Classification

Sukhendra singh[1], B K Tripathi[2]

[1]Department of Information Technology
JSS Academy of Technical Education Noida UP, India
sukhendrasingh@gmail.com

[2] Department of Computer Science and Engineering,
Harcourt Butler Technological University Kanpur, UP India
abkt.iitk@gmail.com

Abstract: **Convolution neural networks (CNN) have shown the state of the art performance for visual recognition, classification of images, time series, and sequential data. Despite good performance, it has few serious drawbacks that it cannot properly encode the orientation and spatial positioning of components of the input data and it also suffers from overfitting. Quaternion CNN is a generalization of traditional CNN and it can properly encode internal and external dependencies between the components of the input data and it is free from overfitting and its generalization performance is better than conventional CNN. It can be modified to work with all Deep Neural Network (DNN) models with quaternion as input. In this paper, the authors have proposed quaternion residual learning for the classification of breast cancer in the BreakHis dataset of breast histopathological images. They have observed that quaternion CNN outperforms to real CNN. The experiment has obtained the classification accuracy of 98.04% and F-score of 0.9842.**

Keywords**:** Deep Neural network, Residual Network, Quaternion neural network, Deep Convolution Neural Network, Histopathological Image Analysis, Deep learning**.**

I. Introduction

Breast Cancer is one of the leading cause of deaths in women across the globe. In this disease, an abnormal growth is observed in breast cells. This abnormal growth makes tumors in breast. These tumors can be benign or malignant. Regular checkups for the early detection is the only cure of this disease and in later stages it becomes very difficult for the doctors to save the patents. Although it is very difficult to diagnose the presence of tumors by visual examination by doctors.Deep learning has shown notable performance for a wide variety of tasks of image recognition, classification, and prediction. In most of the applications, they have outperformed machine learning classifiers using handcrafted feature extraction. It has resulted in many different pre trained architectures AlexNet[1], VGGNet [2], GoogleNet [3] and ResNet[4],[5] etc. All these architectures are constituted of CNN modules with a huge number of layers in between. They have shown surprising results in various visual recognition tasks when they are applied to images and videos. They have also performed excellently for time series and sequential data. The increasing number of layers in between increases accuracy up to a point, beyond this point, accuracy starts decreasing. High dimensional neural networks (HDNN) [6-8] which are inspired by complex and hyper-complex algebra, have drawn the attention of researchers and they have observed that HDNN achieves better accuracy and employs fewer numbers of parameters in comparison to the equivalent real-valued neural network. In HDNN, quaternion-valued network (QVNN), input, weights, biases, and output values are quaternion values. Quaternion CNN has been proposed by redesigning custom CNN modules for compatibility to the quaternion domain. QCNN employs flexible regularization techniques to prevent overfitting. For images, the spatial arrangement of the pixel, and for temporal and sequential data, QCNN better exploit the structure of data, especially when the data is organized in different channels. A neural network is efficient only if it can extract a hidden representation of input data in terms of edges and shapes which further constitute more complex shapes like the nose of a person etc. in a facial image of a person. The neural network should handle global and local dependencies inherent in the input data. In the context of image dataset, global dependency reflects interactions among different pixels in an image while local dependency reflects mapping among three channels (R, G, B) in the images. Conventional CNN only pays attention to global dependency but poorly manages local dependency that is where neural networks based on hyper-complex algebra can also manage interactions within different channels in a pixel. Hyper-complex numbers have many interesting properties which real numbers don't. So the main motivation is to harness the power of complex numbers [9-11],[14-15], quaternion[12], and octonions[13] in deep learning to train a neural network that can classify breast histopathology images with improved accuracy. Residual Network has already given outstanding results for image classification tasks. Quaternion representation [16] of an image enables us to extract more complicated features from a color image. Prediction of breast

cancer from histopathological images is very complicated than predicting cats and dogs from a given input image because the former will require the extraction of more complicated features than the later. An attempt has been made by making design changes in residual networks so that it is compatible with quaternion-valued inputs and delivering better results.

CNN has also given outstanding performance for tumor detection and classification of various diseases by analyzing histopathological images. These models are used by computer-aided detection and diagnostic system (CAD)[17],[18] to assist pathologists for correct prediction and preparation of an accurate report. This CAD system saves a lot of time and effort from pathologists. In this paper, the contribution of authors is to implement a quaternion residual network on BreakHis, the histopathological dataset of breast lesions, and compared with recent work. The main novelty of our work is to extend the residual network in the quaternion domain. This extended new architecture can offer a better structural representation of input where the dependencies and interrelatednesses of multi-channel input are preserved which enables to extract better descriptor of an input object and hence improves performance. The rest of this paper is organized as follows: Section 2 presents the algebra of quaternions and essentials of quaternion convolution network, residual network, and characteristics of used dataset followed by the algorithmic description of the proposed model formulation and performed experimentation. Section 3 describes the obtained results and their comparison with recent similar works. Section 4 concludes the work.

II. Proposed model formulation and preliminaries

We have proposed to build a quaternion residual network for the classification of the histopathological image of breast lesions. We have applied this architecture on the BreakHis dataset of histopathological images of breast lesions.

A. Quaternion Convolution Neural Networks

QCNN is a quaternionic extension of real CNN. Quaternion consists of the 4D vector space with basis 1, i, j, and k. This vector space can be split into two orthogonal subspaces, 1st1-D a scalar subspace and 2nd 3-D a pure subspace. Figure 1 and Figure 2 show the difference between constituent blocks of conventional CNN and quaternion CNN.

A quaternion Q can be shown as

$$Q=r+xi + yj + zk \quad (1)$$

Imaginary components in Quaternion are related as

$$i^2=j^2=k^2=ijk=-1 \quad (2)$$

The multiplication of two quaternions does not show the commutative property.

$$ij=k=-ji \quad jk= -kj=i \quad ki= -ik=j \quad (3)$$

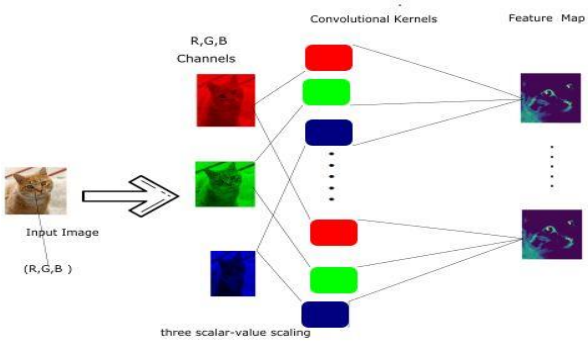


Figure 1. Feature maps in Real CNN

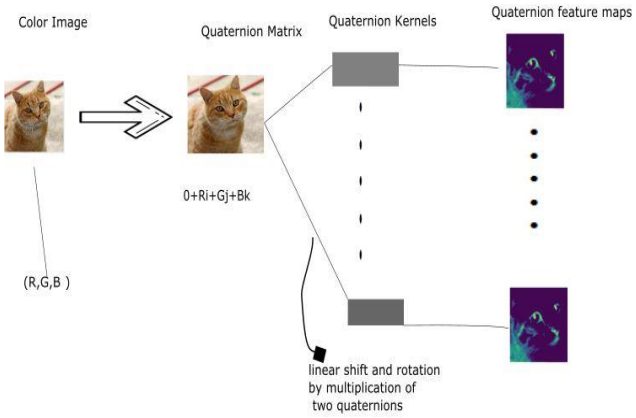


Figure2. Feature maps in QCNN

In a quaternion, r is the scalar part, while xi, yj, and zk are the components of the imaginary part xi + yj + zk, or vector part written as v

$$Q=(r,v) \quad (4)$$

The conjugate Q* of Q is given as :

$$Q^*=(r-xi - yj - zk) \quad (5)$$

The norm of Q written as ||Q|| is given by

$$||Q||=\sqrt{r^2 + x^2 + y^2 + z^2} \quad (6)$$

The inverse Q-1 of Q is described by the formula

$$Q^{-1}=\frac{Q^*}{||Q||^2|} \quad (7)$$

Similar to the complex number, a quaternion number can be shown as

$$Q= \rho e^{\theta s}=\rho(\cos\theta + s \sin \theta) \quad (8)$$

ρ =|Q|, θ is a real number and s is a pure imaginary unit quaternion

On rotation of a 3-D vector Q by an angle θ along a rotation axis w to output a new vector p^. This rotation can be shown with an equation as follows.

$\hat{p}= \widehat{w} . \widetilde{Q}. \widehat{w}$ where $\hat{p}$ and $\widetilde{Q}$ are pure quaternion with zero real part.

$\widetilde{Q}= q_1 i+ q_2 j +q_3 k$ and $\hat{p} = p_1 i +p_2 j+p_3 k$

$$\widehat{w}= \cos\frac{\theta}{2}+\sin\frac{\theta}{2}(w_1+w_2+w_3) \quad (9)$$

Quaternion Convolution operation is performed by scaling and rotation between input Q and quaternion convolution filter.

Applying convolution on quaternion gives a linear combination of each axis as shown in Figure 3.

A color patch in an image can be represented in the form of quaternion matrices in which three channels (Red, Green, Blue) are shown as three imaginary axes.
The patch can be described as pure quaternion with given in Eq.(10)

$$X=0 + x_r i + x_g j + x_b k \quad (10)$$

Figure 3. A color image patch is shown as three axes of quaternion

If w is quaternion filter with size F and Q is quaternion matrix with input size N. Then quaternion convolution operation is described as

$$Q \otimes w = [\widetilde{f_{kk'}}] \in H^{(N-F+1)\times(N-F+1)} \quad (11)$$

$$f_{kk'} = \sum_{l=1}^{M}\sum_{l'=1}^{M} \frac{1}{s_{ll'}} w_{ll'}\, q(k+l)(k'+l')\, \overline{w_{ll'}} \quad (12)$$

$$w_{ll'} = s_{ll'}\left(\cos\frac{\theta_{ll'}}{2} + \mu \sin\frac{\theta_{ll'}}{2}\right) \quad (13)$$

Here θ lies between-π and π and s is scaling factor and μ is unit length axis.
Hamiltonian product [19] enables a quaternion neural network to extract these internal latent relations within the features of the quaternion. This is the main reason for the quaternion neural network performing better than a real-valued neural network.

B. *Hamiltonian Product*

The dot product used in real-valued CNN is substituted by Hamiltonian product in QCNN and it is used to calculate the product of two quaternions.
Let us take two quaternions Q_1 and W_1 as follows
$Q_1 = r_1 + x_1 i + y_1 j + z_1 k$ and $W_1 = r_2 + x_2 i + y_2 j + z_2 k$
The Hamiltonian product ⊗ between Q_1 and W_1 is defined as follows
$Q_1 \otimes W_1 = (r_1r_2 - x_1x_2 - y_1y_2 - z_1z_2) + (r_1x_2 + x_1r_2 + y_1z_2 - z_1y_2)\mathbf{i} + (r_1y_2 - x_1z_2 + y_1r_2 + z_1x_2)\mathbf{j} + (r_1z_2 + x_1y_2 - y_1x_2 + z_1r_2)\mathbf{k}$.
Convolving a quaternion Q_1 by a quaternion filter W_1 outputs a quaternion as follows in Eq. (14):-

$$\mathbf{Q_1 \otimes W_1} = \begin{bmatrix} r_1 & -x_1 & -y_1 & -z_1 \\ x_1 & r_1 & -z_1 & y_1 \\ y_1 & z_1 & r_1 & -x_1 \\ z_1 & -y_1 & x_1 & r_1 \end{bmatrix} * \begin{bmatrix} r_2 \\ x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} r' \\ x'i \\ y'j \\ z'k \end{bmatrix} \quad \mathbf{(14)}$$

The Hamiltonian product enables QCNN to extract local dependency on the features of a quaternion.
The 4 ×4 matrix on the right-hand side of Eq (15) can be written as

$$\begin{bmatrix} r_1 & -x_1 & -y_1 & -z_1 \\ x_1 & r_1 & -z_1 & y_1 \\ y_1 & z_1 & r_1 & -x_1 \\ z_1 & -y_1 & x_1 & r_1 \end{bmatrix} = r1\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + x1\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix} + y1\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix} + z1\begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

(15)
The Hamiltonian product can be shown with Figure 4.

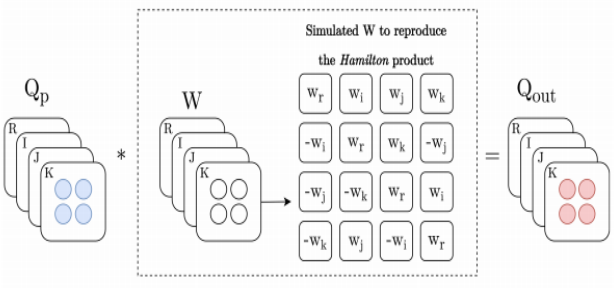


Figure 4: Process of Quaternion Convolution

C. *Quaternion Weight Initialization*

To ensure faster convergence and avoiding the problem of vanishing gradient, suitable parameter initialization needs to be used. In this paper, the same mechanism has been used as that of [20,21]. The components of weight matrix W are w_r, w_i, w_j and w_k can be initialized as
$w_r = \varphi\cos(\theta)$
$w_i = \varphi q_i i \sin(\theta)$
$w_j = \varphi q_i j \sin(\theta)$
$w_k = \varphi q_i k \sin(\theta)$
where θ is chosen randomly from the interval [-π, π] and $q_i = 0+xi+yj+zk$
and xi, yj and zk are generated by uniform distribution from the set [0,1]. φ is generated randomly from the set [-σ, σ] where

$$\sigma = \frac{1}{\sqrt{2(n_{in} + n_{out})}}$$

n_{in} and n_{out} are numbers of neurons in the input layer and output layers.

D. *Quaternion Batch Normalization*

The main objective of batch normalization is fastening the training process and stabilizes it. This has been applied in real CNN but for complex and quaternion CNN, this is more complicated but advantageous. The method used for batch normalization in RCNN cannot be used in QCNN. In [16] the authors have used a matrix whitening approach.
First, a covariance matrix is formed with all covariant values for four components of input quaternions.

If input quaternion $Q = r_1 + x_1 i + y_1 j + z_1 k$ then Covariance matrix V is calculated as in Eq. 9.

$$V = \begin{bmatrix} V_{rr} & V_{ri} & V_{rj} & V_{rk} \\ V_{ir} & V_{ii} & V_{ij} & V_{ik} \\ V_{jr} & V_{ji} & V_{jj} & V_{jk} \\ V_{kr} & V_{ki} & V_{kj} & V_{kk} \end{bmatrix} \quad (16)$$

Where V_{ij} is the covariance between the ith and the jth component of quaternion.
Then a matrix is formed by applying Cholesky decomposition [24] on V^{-1} and then this matrix is employed to whiten the input data.
$\bar{x} = W(x - \mu_x)$ where μ_x is mean.
The quaternion batch normalization can be described with the equation $Bn(\bar{x}) = \gamma\bar{x} + \beta$
γ , β are two learnable parameters where β is an also quaternion and γ is a 4×4 matrix as follows

$$\gamma = \begin{bmatrix} \gamma_{rr} & \gamma_{ri} & \gamma_{rj} & \gamma_{rk} \\ \gamma_{ir} & \gamma_{ii} & \gamma_{ij} & \gamma_{ik} \\ \gamma_{jr} & \gamma_{ji} & \gamma_{jj} & \gamma_{jk} \\ \gamma_{kr} & \gamma_{ki} & \gamma_{kj} & \gamma_{kk} \end{bmatrix} \quad (17)$$

E. Quaternion Fully Connected Layer: this is the layer where actually classification takes place and it maintains more internal relationship information and fetches better descriptors than RCNN. Quaternion FC layer with same shaped kernels the same as inputs. If the input is N-D quaternion

$\hat{a} = [\hat{a_i}] \in H^N$ for i=1,2,3…….N and applying M 1D filter $and\ \widehat{w^m} = [\hat{w_i^m}] \in H^M$.

To calculate output $\hat{b} = [\widehat{b_m}] \in H^M$. The following equation can be used.

$$b_m = \sum_{i=1}^{N} \frac{w_i^m \, a_i \, w_i^{m*}}{s_i} \quad (18)$$

here s_i stands for w_i^m .

F. Deep Residual network

As the number of layers on CNN keeps increasing, the network becomes more robust and training accuracy increases. But this happens up to a certain point only. If the number of layers is increased beyond this point, a saturation is witnessed in training accuracy and beyond this point, training accuracy decreases along with testing accuracy with the increase in the number of layers as shown in Figure 5. This is known as a degradation problem. By increasing more number of layers, the error gradient will become smaller and smaller as it moves back to the initial layer and weights will not be properly updated that is why training accuracy decreases as opposite to the intuition. This is known as the vanishing gradient problem. To address both the problems of degradation and vanishing gradient, a deep residual network was proposed in [4], [5].

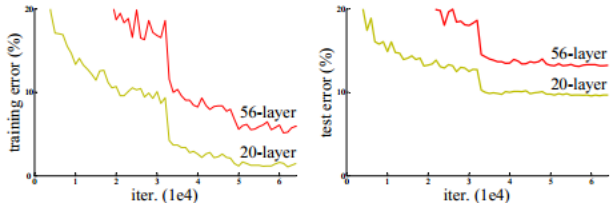


Figure 5: Increase in training error along with a testing error by an increasing number of layers.

The residual network is built based on two principles.
Modularity: One can create a deeper network by repeating the element module from a smaller network to gain increased accuracy.
Residual Block: Figure 6 shows residual block which repeatedly used to build a deep residual network by employing a double or triple layer skip connection to provide an optional path for the gradient. These layers have Relu and batch normalization in between. These skip connections help in overcoming the problem of vanishing gradient.

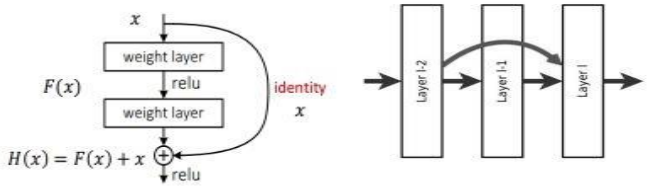


Figure 6: The Residual Block

This network helps to know the appropriate depth of the network. Because with fewer layers, the system will not be trained properly as it will be able to learn enough representation power of input data, and with more number of layers system will face degradation problems. One advantage of the residual network is that even if more layers are there in the network then the extra layers will learn to act as identity function without degrading the performance. The paper [4] has shown that deep residual network has outperformed the networks like AlexNet, VGGNet by introducing skip connections in the architecture.
Characteristics of used Data set: The authors have experimented on a BreakHis [22],[23], a dataset of breast histopathological images. Many researchers have performed their experiments on this dataset. There are a total of 7909 histopathology images that were collected from 82 patients. These images have been taken with different magnifying factors 40X, 100X, 200X, and 400X. The P&D Laboratory have collaborated to build this dataset. The images in the dataset are two classes: benign and malignant tumors. Benign tumors are not cancerous but malignant tumors are cancerous and a matter of serious concern. Benign tumors are localized and do not spread to the neighboring tissues and malignant tumors are invasive and they spread to nearby cells. The images for this dataset have collected by surgical open biopsy (SOB) method. The benign and malignant tumors are further categorized into four groups based on the characteristics of tumor cells. The subtype of breast tumors are adenosis (A), fibroadenoma (F), phyllodes tumor (PT), and tubular adenoma (TA) and the subtype of malignant tumors are carcinoma (DC), lobular carcinoma (LC), mucinous carcinoma (MC) and papillary carcinoma (PC).

MAGNIFICATION	BENIGN	MALIGNANT	TOTAL
40X	652	1,370	1,995
100X	644	1,437	2,081
200X	623	1,390	2,013
400X	588	1,232	1,820
TOTAL	2,480	5429	7909
PATIENTS	24	58	82

Table 1 Describes the structure of the BreakHis dataset

Figure 7 shows malignant tumor structural variations of different magnification.

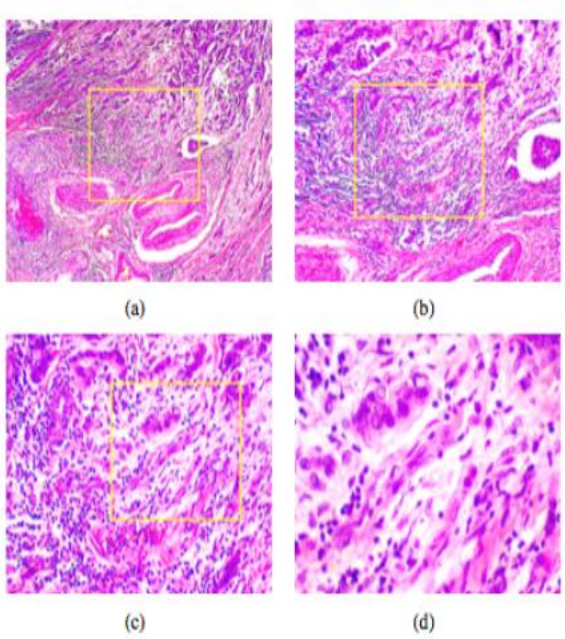


Figure 7: Malignant tumors images with different magnification (a) 40X, (b) 100X, (c) 200X, and (d) 400X.

Benign Class subtype structure: Table 2 shows the structure of subtypes.

MAGNIFICATION	A	F	TA	PT	TOTAL
40X	114	253	109	149	1995
100X	113	260	121	150	2081
200X	111	264	108	140	2013
400X	106	237	115	130	1820
TOTAL	444	1014	453	569	2368
# PATIENTS	4	10	3	7	24

Table 2: Benign class subtype structure

Malignant Class subtype structure: Table III shows the structure of subtypes.

MAGNIFICATION	DC	MC	LC	PC	TOTAL
40X	864	156	205	145	1370
100X	903	170	222	142	2437
200X	896	163	196	135	1390
400X	788	137	169	138	1232
TOTAL	3451	626	792	560	5429
# PATIENTS	38	5	9	6	58

Table 3: Malignant tumor subtypes

III. Proposed Architecture: The authors have proposed a method by applying a quaternion residual network on the image set. The first, the dataset was prepared and then it was converted into quaternion and then it was input to quaternion residual network by customizing convolution layer and fully connected layer to make it compatible for a quaternion. Being color images, so images were intrinsically thought to be used as a quaternion. Figure 8 shows our proposed architecture.

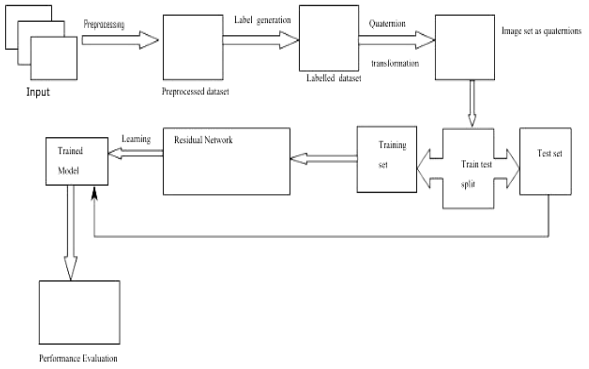


Figure 8: System Architecture of Quaternion residual network

Algorithm: Input= Dataset Output=Trained Model, performance on D_{test}

1. for each image i in D dataset:
   Preprocess(i) // normalize each image to the same size
   LabelGeneration(i) // labels are either malignant
   RGBtoQuaternion(i) //each image is converted into //quaternion matrix
2. TrainTestSplit(split_ratio)// for a given split ratio D is //divided into Dtrain and Dtest
3. Model=QuaternionResidualNetwork(D_{train})
4. Performance=evaluate(Model, D_{test})

QuaternionResidualNetwork(Dtrain):

1. For each image i in Qtrain:
   QConv2D(I,kernel_size,stride,padding)
   BatchNormaization()
   RELU()
   for j=1 to N: //N is number of residual blocks
   MaxPooling()
   ResidualBlock()
   GlobalAveragePooling()
   Dense(relu)
   Dropout(dropout_ratio)
   Model=Dense(sigmoid)
   Performance=

Algorithm for Residual Block:

ResidualBlock(in):

For k=0 to N:
QConv2D(in)
batchNormalization()
res=ELU()
res=res+in

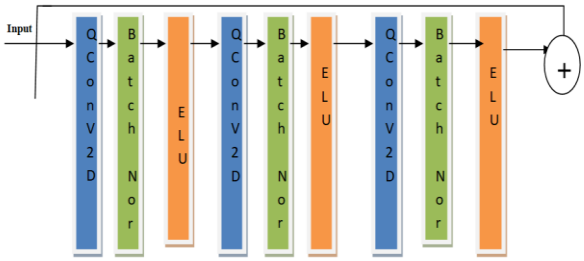


Figure 9: Structure of Quaternion Residual Block

The internal structure of used quaternion residual network: In Figure 10, the internal structure of the deep residual network used in the architecture is shown. In this network, the input is in the form of a quaternion matrix. In this architecture, four residual blocks have been taken. Max pooling block reduces the dimensionality of feature maps obtained after each convolution layer by ensuring that less trainable parameters. Global average pooling has been deployed to minimize over-fitting and it gives 1D vector which is inp*u*t to the FC layer.

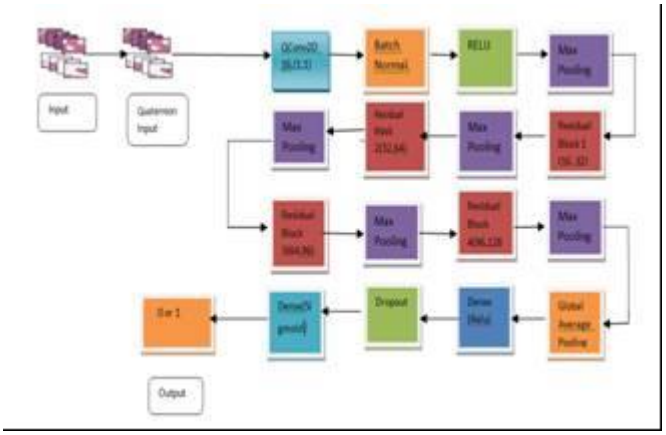


Figure 10: Structure of Quaternion Residual network

Performance Evaluation: Finally the trained model from the proposed architecture is tested against test-set and various performance metrics like accuracy, sensitivity, specificity, F-score, AUC are computed. The confusion matrix was also evaluated for the model.

IV. Implementation and Results

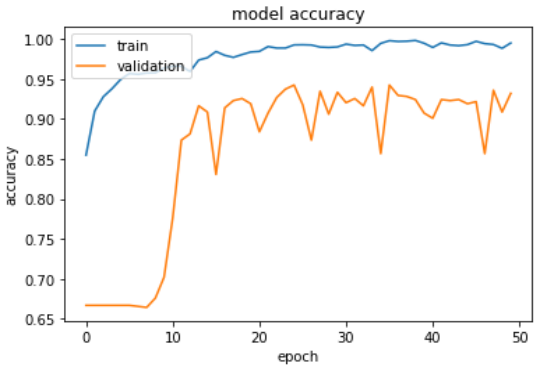
This architecture was implemented in Python. We have worked in 16GB RAM and a single 12GB NVIDIA Tesla K80 GPU. The size of each image in the dataset is 700×460. For each split ratio of 20%, 30%, and 40%, the same experiment was done 10 times for each split ratio and average values of all performance metrics are being reported here. Each image in the dataset was resized to 50×50. Keras framework was used to build this architecture. The parameters were optimized with adam optimizer with .001 as the learning rate. Optimized weights were computed using early stopping techniques. There are a total of 7909 images. Table IV displays the size of training data and testing data for each split ratio.
Figure 10 shows the structure of our proposed model with parameters being used in the implementation.

SPLIT RATIO	TRAIN-DATA	TEST-DATA	ACCURACY	SENSITIVITY	SPECIFICITY	F-SCORE	AUC
.2	6327	1582	98.04	.9868	.9552	.9842	.99
.3	5539	2370	94.14	.9529	.9166	.9559	.98
.4	4745	3164	91.40	.9398	.8051	.9197	.96

Table 4: Performance of the proposed model

Table 4 shows the performance of the proposed model on the BreakHis dataset.

In the proposed architecture, there were a total of 569,345 parameters out of which 560,769 were trainable and the remaining 8576 were un-trainable. The remaining plots are



for a split ratio 20%.

Figure 11: Accuracy plot

We have applied equivalent real valued same network on same dataset and all parameters being same and we have found out that in the case of quaternion valued neural network it took less number of epochs to converge and because of Hamiltonian product in Quaternion networks in place of dot product in real valued network, it has shown better classification test accuracy our breast cancer classification task. Although training time was much greater in comparison to real valued network but it can be sacrificed to get more accuracy which is more important than time in our current task of cancer classification.

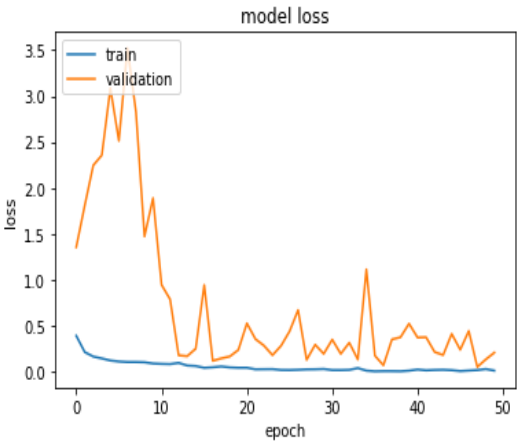


Figure 12: Train-validation loss curve

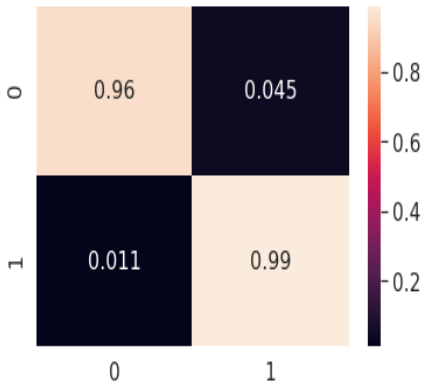


Figure 13: Confusion matrix

Reference	Year	Dataset	Approach	Accuracy	F Score
[25] A. Cruz-Roa et al.	2015	In-house	3-layered CNN	84.23	71.28
[26] M. Balazsi et al.	2016	In-house	Slic registration technique [27] and random forest for classification	88.7%,	79.50
[28] Gallego-Posada et al.	2016	Mini MIAS[29]	CNN (Alexnet and VGG SVM	64.52	76.80
[30] B. E. Bejnordi et al.	2017	Breast histopathology images from kaggle repository	Two stacked -CNN architecture	81.3%	79.60
[31] S. Reza and J. Ma	2018	Breast histopathology images from kaggle repository	Regularized CNN dealing with class imbalance dataset	84.44	81.36
[32] A. Nahid et al.	2018	Breakhis	Cnn and lstm	90%	93
Our Proposed work	-	Breakhis	Quaternion residual network	98.04	98.42

Table 5: Comparison with Baseline Performance

V. Conclusion

QCNN have been applied for speech classification, natural language processing with remarkable performance. High dimensional deep learning classifiers can further be used many computer vision tasks with better performance than its real counterpart. In this paper, we have applied quaternion convolution network to build a residual network for classification of BreakHis dataset of breast histopathology images into two classes of benign and malignant. Since images were represented as a quaternion, it enabled the extraction of better representational features which were further classified by the deep residual network. The experiment has delivered the test accuracy of 98.04, sensitivity .9868, and specificity .9552. This architecture required less parameter to train in comparison to real-valued residual network producing the same accuracy. The main challenge was the lack of publically available large annotated dataset of histopathology images. Histopathology images in the dataset should be collected from a large number of patients. Hyper-complex algebra has a lot of merits over real-valued algebra; it needs to be plugged with the new and faster deep learning architectures for speech recognition and classification, image and video classification, image segmentation.

Figure 11 and 12 display the accuracy and loss curve. with respect to epochs. Figure 13 and 14 show the confusion matrix and ROC curve of the proposed architecture.

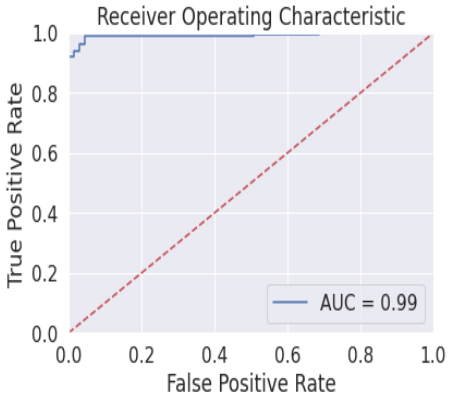


Figure 14: ROC Curve

Reference

[1] Almisreb, A. A., Jamil, N. & Din, N. M. "Utilizing AlexNet Deep Transfer Learning for Ear Recognition". in 4th International Conference on *Information Retrieval and Knowledge Management: Diving into Data Sciences*, CAMP, 2018.

[2] Wang, L., Guo, S., Huang, W. & Qiao, Y. "Places205-VGGNet Models for Scene Recognition", 12–13,2015.

[3] Ballester, P. & Araujo, R. M. "On the performance of googlenet and alexnet applied to sketches". in 30th *AAAI Conference on Artificial Intelligence*, 1124–1128 ,2016.

[4] He, K., Zhang, X., Ren, S. & Sun, J. "Deep Residual Learning for Image Recognition". in Proceedings of *the IEEE conference on computer vision and pattern recognition* 770–778 ,2016.

[5] Zagoruyko, S. & Komodakis, N. "Wide Residual Networks" . in British Machine Vision Conference 2016, BMVC 2016 87.1-87.12 ,2016.

[6] Tripathi, B. K. “High Dimensional Neurocomputing” *Stud. Comput. Intell*. 571, 79–103 ,2015.
[7] Tripathi, B. K. & Kalra, P. K. “On efficient learning machine with root-power mean neuron in complex domain”. *IEEE Trans. Neural Networks* 22, 727–738 ,2011.
[8] Tripathi, B. K. & Kalra, P. K. “High dimensional neural networks and applications”. *Stud. Comput. Intell*. 275, 215–233 ,2010.
[9] Guberman, N. “On Complex Valued Convolutional Neural Networks”. (The Hebrew University of Jerusalem Israel, 2016).
[10] Nitta, T. “Orthogonality of Decision Boundaries in Complex-Valued Neural Networks”. *Neural Comput*. 16, 2004.
[11] Nitta, T. “Three-Dimensional Vector Valued Neural Network and its Generalization Ability”. 10, 237–242 ,2006.
[12] Isokawa, T., Kusakabe, T., Matsui, N. & Peper, F. “Quaternion neural network and its application”. in International conference on *knowledge-based and intelligent information and engineering systems* vol. 2774 318–324, 2003.
[13] Wu, J. et al. “Deep octonion networks”. *Neurocomputing* 397, 179–191 ,2020.
[14] Trabelsi, C. et al. “Deep Complex Networks”. 1–19 2017.
[15 Popa, C. A. “Complex-Valued Deep Belief Networks”. in *Lect. Notes Comput. Sci* (ed. T. Huang) 72–78 ,2018.
[16] Parcollet, T., Morchid, M. & Linares, G. “Deep quaternion neural networks for spoken language understanding”. in 2017 IEEE *Automatic Speech Recognition and Understanding Workshop*, 504–511 2018.
[17] Tang, J., Member, S., Rangayyan, R. M., Xu, J. & Naqa, I. El. “Computer-Aided Detection and Diagnosis of Breast Cancer With Mammography : Recent Advances”. *IEEE Trans. Inf. Technol. Biomed.* 13, 236–251.2009
[18] Alazab, M. et al. “COVID-19 prediction and detection using deep learning”. *Int. J. Comput. Inf. Syst. Ind. Manag. Appl*. 12, 168–181,2020).
[19] Parcollet, T., Morchid, M. & Linarès, G. “A survey of quaternion neural networks”. *Artif. Intell. Rev*. 53, 2957–2982 ,2020.
[20] Parcollet, T., Morchid, M., Bousquet, P., Dufour, R. & Linar, G. “QUATERNION NEURAL NETWORKS FOR SPOKEN LANGUAGE UNDERSTANDING”. *IEEE Spoken Language Technology Workshop*. IEEE, 2016.
[21] Parcollet, T. et al. “Quaternion Convolutional Neural Networks for End-to-End Automatic Speech Recognition.” arXiv preprint arXiv:1806.07789. 2018 .
[22] Gaudet, C. J. & Maida, A. S. “Deep Quaternion Networks”. *Int. Jt. Conf. Neural Networks* 1–8 ,2018.
[23] Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. “A Dataset for Breast Cancer Histopathological Image Classification”. IEEE Trans. Biomed. Eng. 63, 1455–1462 ,2016.
[24] Benhammou, Y., Achchab, B., Herrera, F. & Tabik, S. “BreakHis based breast cancer automatic diagnosis using deep learning: Taxonomy, survey and insights”. *Neurocomputing* ,375, 9–24 ,2020.
[25] Cruz-roa, A., Basavanhally, A., Gonz, F., Gilmore, H. & Feldman, M. “Automatic detection of invasive ductal carcinoma in whole slide images with Convolutional Neural Networks”. 9041, 1–15 ,2014.
[26] Balazsi, Matthew, Paula Blanco, Pablo Zoroquiain, Martin D. Levine, and Miguel N. Burnier Jr. "Invasive ductal breast carcinoma detector that is robust to image magnification in whole digital slides." *Journal of Medical Imaging* 3, no. 2 ,2016.
[27] Achanta, Radhakrishna, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. "SLIC superpixels compared to state-of-the-art superpixel methods." *IEEE transactions on pattern analysis and machine intelligence* 34, no. 11 (2012):
[28] J. D. Gallego-Posada, : D. A. Montoya-Zapata, § and O. L. Quintero-Montoya. “Detection and Diagnosis of Breast Tumors using Deep Convolutional Neural Networks”. In Conference Proceedings of XVII *Latin American Conference in Automatic Control*, vol. 17. 2016. 1–9 ,2011.
[29] Suckling, J. et al. “The mini-MIAS database of mammograms. The Mammographic Image Analysis Society Digital Mammogram Database Exerpta” *Medica. International Congress Series* 1069 (1994).
[30] Bejnordi, B. E. et al. “Context-aware stacked convolutional neural networks for classification of breast carcinomas in whole-slide histopathology images”. *Journal of Medical Imaging* , no. 4 (2017).
[31] Reza, M. S., & Ma, J. “Imbalanced histopathological breast cancer image classification with convolutional neural network”. In 14th *IEEE International Conference on Signal Processing* ,pp. 619-624,2018.
[32] Nahid, A. A., Mehrabi, M. A., & Kong, Y. “Histopathological breast cancer image classification by deep neural network techniques guided by local clustering”. *BioMed research international*, 2018.

Author Biographies

Sukhendra Singh is pursuing his PhD degree in computational intelligence area from Dr. APJ Abdul Kalam Technical University, Lucknow, India, and completed his M.Tech degree in Computer Science from Birla Institute of Technology Mesra, Rachi and B.Tech in Information technology from Indian Institute of Technology Allahabad, India. He is currently an Assistant Professor in Department of Information Technology of JSS Academy of Technical Education Noida, India. His areas of research include, , machine learning and Computer vision. Email: sukhendrasingh@gmail.com

Bipin Kumar Tripathi completed his PhD degree in computational intelligence from IIT Kanpur, India, and M.Tech degree in computer science and engineering from IIT Delhi, India. Dr.Tripathi is currently serving as a professor in Department of Computer Science and Engineering of HBTU Kanpur,India. He is also leading the Nature inspired Computational Intelligence Research Group (NCIRG)at HBTI. His areas of research include high-dimensional neurocomputing, computational neuroscience, intelligent system design, Machine learning and computer vision focused on biometrics, and 3D imaging. **Email: abkt.iitk@gmail.com**