Multimedia Tools and Applications (2022) 81:1743–1764
https://doi.org/10.1007/s11042-021-11409-7

Pneumonia classification using quaternion deep learning

Sukhendra Singh[1] · **B. K. Tripathi**[2]

Received: 1 November 2020 / Revised: 20 January 2021 / Accepted: 2 August 2021 /
Published online: 12 October 2021

Abstract

What is pneumonia

Pneumonia is an infection in one or both the lungs because of virus or bacteria through breathing air. It inflames air sacs in lungs which fill with fluid which further leads to problems in respiration. Pneumonia is interpreted by radiologists by observing abnormality in lungs in case of fluid in Chest X-Rays. Computer Aided Detection Diagnosis (CAD) tools can assist radiologists by improving their diagnostic accuracy. Such CAD tools use neural networks which are trained on Chest X-Ray dataset to classify a Chest X-Ray into normal or infected with Pneumonia. Convolution neural networks have shown remarkable performance in object detection in an image. Quaternion Convolution neural network (QCNN) is a generalization of conventional convolution neural networks. QCNN treats all three channels (R, G, B) of color image as a single unit and it extracts better representative features and which further improves classification. In this paper, we have trained Quaternion residual network on a publicly available large Chest X-Ray dataset on Kaggle repository and obtained classification accuracy of 93.75% and F-score of .94. We have also compared our performance with other CNN architectures. We found that classification accuracy was higher with Quaternion Residual network when we compared it with a real valued Residual network.

How it is interpreted ?

which tools are used ?

what these tools internally use?

what is qcnn?

which one is specialised ?

which one is better?

Keywords Deep learning · Convolution neural network · Computer aided detection and diagnosis · Quaternion convolution neural network · Residual network · High dimensional neural network

1 Introduction

what type of disease is pneumonia ?

how it is caused?

Pneumonia is one of the thoracic diseases that are caused by infection in lungs because of spread of virus, bacteria and fungi through breathing air in lungs. This infection gets fluid in lungs and that further creates respiratory problems. Pneumonia is one of the biggest

How does it gets infected?

✉ Sukhendra Singh
sukhendrasingh@gmail.com

B. K. Tripathi
abkt.iitk@gmail.com

[1] JSS Academy of Technical Education, Noida, India

[2] Harcourt Butler Technological University Kanpur, Kanpur, India

why are we choosing this statement as ?

causes of hospitalization and death especially in young children. Chest X-Ray (CXR) is the most effective method to diagnose Pneumonia as it is more widespread and economical than other conventional tests.CXR are carefully analyzed and interpreted by expert clinicians. This is a very time consuming task and sometimes this results in disagreement among various radiologists for prediction CXR. This disagreement can be eliminated by deploying deep learning powered solutions which automates the task of detecting abnormality in CXR images with high accuracy. These CAD systems reduce radiologists' efforts and increase reliability of their reports.

In past studies, researchers had employed hand crafted feature extraction methods for transforming raw images into useful features for classifying a medical image for disease classification. This requires domain expertise for performing feature engineering by extracting relevant features, transforming and removing redundant features. More promising results have been shown by deep neural network approaches where feature engineering has been replaced by feature learning where better features are learnt. In the last few layers of the architecture, very complex features are extracted from input images which are further classified.

Figure 2 shows two chest X-Rays images one belongs to a healthy person, no abnormality observed in chest X-Rays image and the other one shows the presence of fluid surrounding the lung, showing presence of Pneumonia. In this paper, we have applied a quaternion valued residual network with similar custom changes so that it is applicable to quaternion valued inputs. We have applied this quaternion residual network architecture for detection and classification of Pneumonia on chest X-Ray image dataset available on Kaggle repository (https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia).

Need of this architecture: In our paper, by making architectural changes, we have extended residual networks [14] to quaternion domain. Color image being a three dimensional entity, it had the natural propensity of applying QCNN to extract most representative feature descriptors. This Quaternion residual network architecture offers a better structural

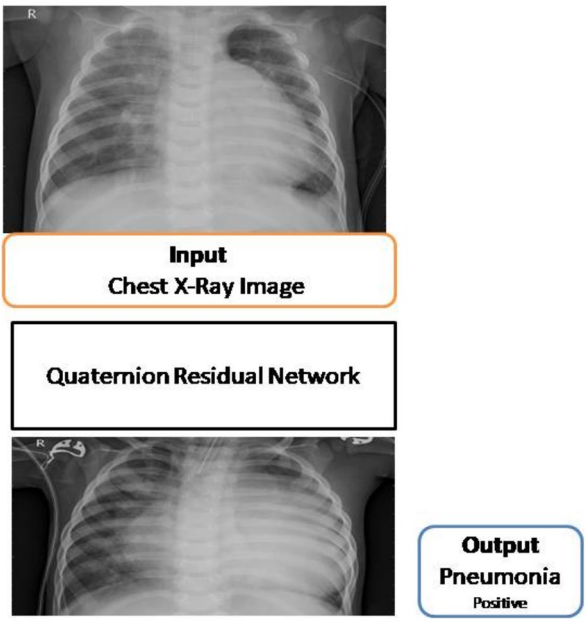


Fig. 1 Zero level diagram of proposed architecture

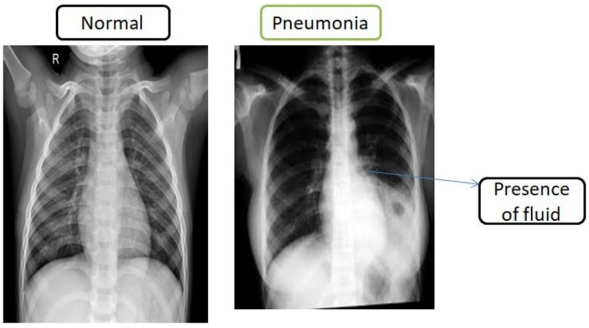


Fig. 2 Difference in Chest X Ray Images in Normal and Pneumonia

representation of input where the dependencies and interrelatedness of multi-channel input are preserved which enables to extract a better descriptor of an input object and hence improves performance. It is also argued that any architecture can be transformed into its deep hyper complex counterpart and it has given improved classification performance.

Organization of the paper: Sect. 2 presents preliminary concepts for Quaternion CNN architecture and related studies, Sect. 3 describes dataset characteristics and model formulation, Sect. 4 describes experimentation and result discussion, Sect. 5 presents conclusion.

2 Background and related works

Deep neural networks [31] have shown the state of the art performances for many different tasks which requires human intelligence. Clinical practices have also been influenced by its impact. Deep learning over the period of time has evolved in the form of a number of architectures of convolution neural networks which can extract features in an object specially an image or a video. A color image is a multidimensional entity in the form of three channels (R, G, and B). When real valued CNN is applied on a color image, it is applied independently channel wise and then obtained features are appended before fully connected layers for classification. Therefore real valued CNN's are unable to encode the relationship among the three channels and hence causes the information loss which is hindrance in obtaining more accuracy. High dimensional neural networks (HDNN) [32, 33] have been extended from real valued neural networks by applying the concepts of complex and hyper complex algebra. Complex valued Neural network [3, 15] has emerged as a stronger substitute to conventional neural networks. Quaternion Convolution neural network was proposed in [40], and it has also been applied with significant improvement in performance for speech recognition [24], human motion recognition [5], and detection of 3-d sound events [7], heterogeneous image processing [25], and color image classification [21]. CNN architectures extended into Quaternion CNN on the basis of quaternion algebra [36] where the Hamiltonian product is used instead of dot product and it injects prior information on the structure of the data, thus capturing the internal relations within multidimensional entities. It has been shown in [17, 20, 26] that classification accuracy is significantly improved when QNN is chosen instead of conventional neural networks and also the former requires less parameters if it is compared with equivalent real conventional neural networks. In QCNN, the Hamilton product acts as a regularizer in a sense and which further prevents overfitting.

what is qcnn ?

Any deep learning architecture can be extended to its quaternion domain by customizing convolution operations, batch normalization, max pooling to quaternion valued instead real values.

2.1 Quaternion convolution neural network

QCNN [11, 40] is a quaternion 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, 1st 1-D a scalar subspace and 2nd 3-D a pure subspace.

Figures 3 and 4 show the difference between constituent blocks of conventional CNN and quaternion CNN.

A quaternion Q can be shown as.

$$Q = r + xi + yj + zk \tag{1}$$

Imaginary components in Quaternion are related as

$$i_2 = j_2 = k_2 = ijk = -1 \tag{2}$$

The multiplication of two quaternions does not show the commutative property

$$ij = k = -ji \; jk = -kj = i \quad ki = -ik = j \tag{3}$$

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

$$Q = (r, v) \tag{4}$$

The conjugate Q* of Q is given as:

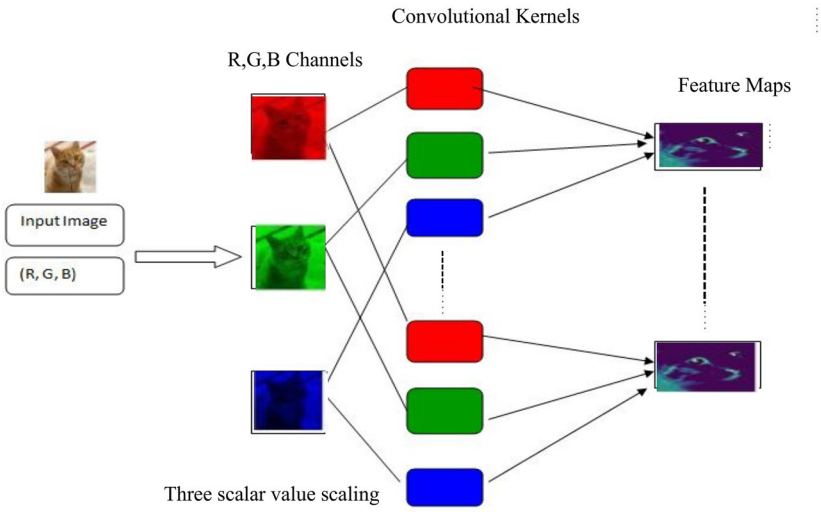


Fig. 3 Feature maps in Real CNN

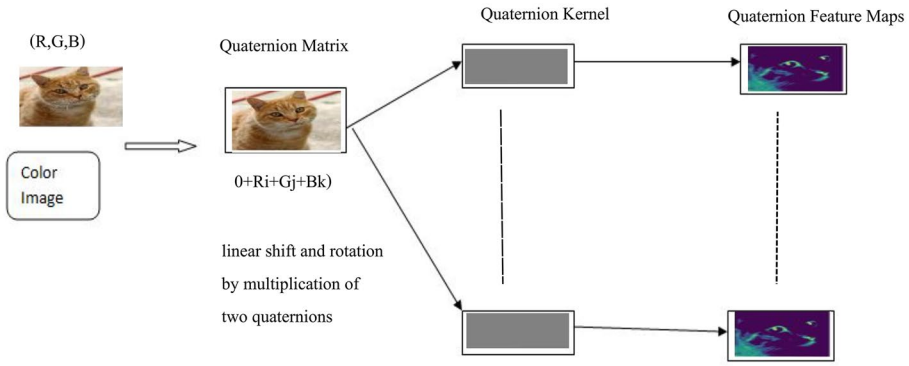


Fig. 4 Feature maps in QCNN

$$Q^* = (r - xi - yj - zk) \tag{5}$$

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

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

The inverse Q^{-1} of Q is described by the formula

$$Q^{-1} = \frac{Q\ *}{||Q||^2|} \tag{7}$$

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

$$Q = \rho e^{\theta s} = \rho(\cos\,\theta + s\,\sin\,\theta| \tag{8}$$

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 $\hat{p}$. This rotation can be shown with an Eq. (9)

$$\hat{p} = \hat{w}.\hat{Q}.\hat{w} \tag{9}$$

Where $\hat{p}$ and $\hat{Q}$ are pure quaternions with zero real part.

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

$\hat{w}$= $\cos\frac{\theta}{2}$ + $\sin\frac{\theta}{2}(w_1 + w_2 + w_3)$.

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 Fig. 5.

A color patch in an image can be represented in the form of quaternion matrices in which three channels (Red, Green, and Blue) are shown as three imaginary axes.

The patch can be described as pure quaternion with given in Eq. (10)

$$X = 0 + x_r i + x_g j + x_b k \tag{10}$$

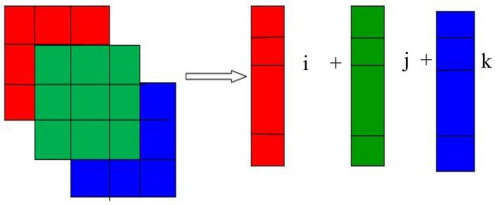


Fig. 5 A color image patch is shown as three axes of pure quaternion

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

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

$$f'_{kk} = \sum \tag{12}$$

$$w_{ll'} = s_{ll'}(\cos\frac{0_{ll'}}{2} + \mu \sin\frac{0_{ll'}}{2}) \tag{13}$$

Here θ lies between-π and π and s is the scaling factor and μ is the unit length axis.

Hamiltonian product enables a quaternion neural network [11] 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.

2.2 Hamiltonian Product

The dot product used in real-valued CNN is substituted by the 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 $\otimes$ between Q_1 and W_1 is defined as follows.

$Q_1 \otimes W_1 = (r_1 r_2 - x_1 x_2 - y_1 y_2 - z_1 z_2) + (r_1 x_2 + x_1 r_2 + y_1 z_2 - z_1 y_2) i + (r_1 y_2 - x_1 z_2 + y_1 r_2 + z_1 x_2) j + (r_1 z_2 + x_1 y_2 - y_1 x_2 + z_1 r_2) k$.

Convolving a quaternion Q_1 by a quaternion filter W_1 outputs a quaternion as follows in Eq. (14):

$$Q_1 \otimes W1 = \begin{bmatrix}\begin{bmatrix} r_1 & -x_1 & -y_1 & -z_1 \\ X_1 & r_1 & -z_1 & y_1 \\ y_1 & z_1 & r_1 & -x_1 \\ z_1 & -y_1 & x_1 & r_1 \end{bmatrix}\end{bmatrix} * \begin{bmatrix}\begin{bmatrix} \boldsymbol{r_2} \\ \boldsymbol{x_2} \\ \boldsymbol{y_2} \\ \boldsymbol{z_2} \end{bmatrix}\end{bmatrix} = \begin{bmatrix}\begin{bmatrix} \boldsymbol{r'} \\ \boldsymbol{x'i} \\ \boldsymbol{y'j} \\ \boldsymbol{z'k} \end{bmatrix}\end{bmatrix} \tag{14}$$

The Hamiltonian product enables QCNN to extract local dependency on the features of a quaternion.

$$\begin{bmatrix}\begin{bmatrix} r_1 & -x_1 & -y_1 & -z_1 \\ X_1 & r_1 & -z_1 & y_1 \\ y_1 & z_1 & r_1 & -x_1 \\ z_1 & -y_1 & x_1 & r_1 \end{bmatrix}\end{bmatrix} = r_1\begin{bmatrix}\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}\end{bmatrix} + x_1\begin{bmatrix}\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \end{bmatrix}\end{bmatrix} + y_1\begin{bmatrix}\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{bmatrix}\end{bmatrix} + z_1\begin{bmatrix}\begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}\end{bmatrix} \tag{15}$$

The 4×4 matrix on the right-hand side of Eq. (15) can be written as.
The Hamiltonian product can be shown with Fig. 6.

2.2.1 Quaternion Weight Initialization

To ensure faster convergence and avoiding the problem of vanishing gradient [13], suitable parameter initialization needs to be used. In this paper, the same mechanism has been used as that of [24]. 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.

2.2.2 Quaternion Batch Normalization

The main objective of batch normalization is fastening the training process and stabilizing it. This has been applied in real CNN but for complex and quaternion CNN, this is more

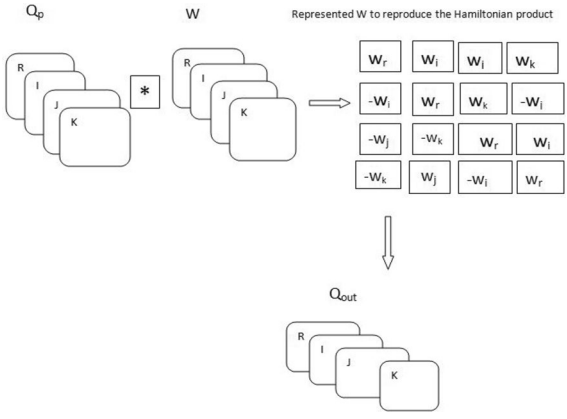


Fig. 6 Process of Quaternion Convolution

complicated but advantageous. The method used for batch normalization in RCNN cannot be used in QCNN [39]. has 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.

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

Then a matrix is formed by applying Cholesky decomposition [11] 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.

$\bar{x} = W(x - \mu_x)$.

γ, β are two learnable parameters where β is an also quaternion and γ is a 4×4 matrix as follows.

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

2.2.3 Quaternion Fully Connected Layer

This is the layer where actual classification takes place and it maintains more internal relationship information and fetches better descriptors than real valued CNN. Quaternion fully connected 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 $and \widehat{w^m} = \left[\hat{w}_i^m\right] \in H^M$

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

$$b_m = \sum_{i=1}^{N} \frac{w_{\hat{i}}^m \, a_i \, w_{\hat{i}}^{m*}}{s_i} \tag{18}$$

here s_i stands for $w_{\hat{i}}^m$.

2.3 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, 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. This is known as a degradation problem. By increasing the 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 opposed 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 [26].

The residual network is based on two principles.

2.3.1 Modularity

One can create a deeper network by repeating the element module from a smaller network to gain increased accuracy.

2.3.2 Residual Block

Figure 7 shows a residual block which is 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.

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 layers in the system will face degradation problems. One advantage of the residual network is that even if mour experimentation, we have got the dataset of Chest X-ore layers are there in the network then the extra layers will learn to act as identity functions without degrading the performance. The paper has shown that deep residual networks has outperformed networks like VGGNet [30] by introducing skip connections in the architecture.

2.4 Related Works

In this section in Table 1, we are summarizing the results of previous studies in related problems.

3 Model Formulation and Proposed Architecture

3.1 Data set Characteristics

For our experimentation, we have got the dataset of Chest X-Ray [19] from Kaggle repository. The images in the dataset are either normal or having Pneumonia. Table 2 presents

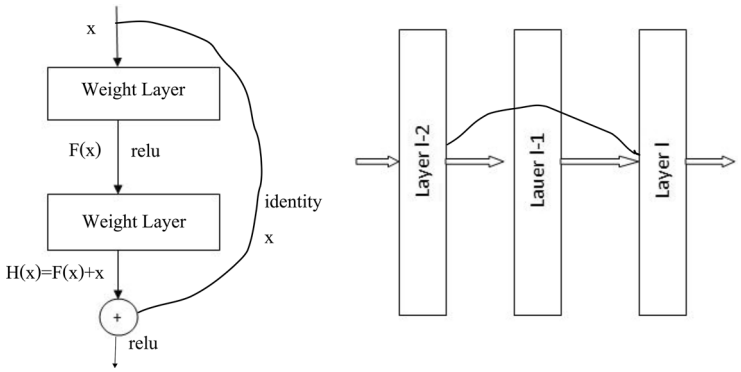


Fig. 7 The Residual Block [14]

Table 1 Recent Studies on Pneumonia detection with their findings

Study	Dataset description	Methods	Results
CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning [27]	They used ChestX-ray14 dataset. It is collection of 112,120 frontal chest X-Rays from 30,805 unique patients with 14 thoracic diseases including Pneumonia. Training set had 98,637 images, validation set had 6351 images, and test set had 420 images and they ensured that there is no data leakage between train set and test set	The authors have used a 121-layered CNN model, CheXNet, trained on the ChestX-ray14. To compare their performance with experienced radiologists, they got their test data annotated by radiologists and evaluated F1 score of their proposed model along with that of the radiologists as well	Their CheXNet model predicted Pneumonia positive with 85% accuracy. The authors found that the CheXNet model obtained a higher average F1 score than the radiologists. The limitation of their proposed work was that only frontal radiographs were examined and historical records of patients were not taken into analysis
Deep Learning for Automatic Pneumonia Detection [10]	The dataset had frontal-view chest-X-ray images obtained from 26,684 unique patients. Images were labeled in either of three classes"Normal","No Lung Opacity / Not Normal","Lung Opacity". The dataset was balanced	The authors have proposed a single model, ensembled over several checkpoints and 4 folds. The model utilizes an SSD RetinaNet with SE-ResNext101 encoder pre-trained on ImageNet. They also used learning rate scheduler available in Pytorch ReduceLROnPlateau with	The results of detection models can change significantly between epochs and depend largely on thresholds. They got ensemble models from different checkpoints to achieve a more stable and reliable solution. The outputs from the same model for 4 cross-validation folds and several check-points were combined before applying non-maximum suppression(NMS) algorithms and optimizing thresholds. Test set, mean average precision (mAP) results were found as 0.24781

Table 1 (continued)

Study	Dataset description	Methods	Results
MultiCheXNet: A Multi-Task Learning Deep Network For Pneumonia-like Diseases Diagnosis From X-ray Scans [9]	Multiple dataset were used for the training process, NIH_chest_xrays, RSNA, SIIM-ACR for classification, localization and segmentation	The authors have applied transfer learning to unseen Pneumonia- like diseases, from the multi-task learning model architecture. Using the common convolution encoder and the classification head, the model weights are transferred, and fine tuned the top layers to classify new unseen, Pneumonia-like diseases. During training, they have applied teacher forcing technique, where they feed ground truth positive classes with certain percent, and classified samples from the first stage classifier for the remaining percent of training samples	For multi task learning with 100% teacher forcing, with the dataset RSNA + SIIM-ACR, it offered 77% accuracy and F1 score 0.71 for classification, dice score as 0.78 for segmentation,0.19 as mean average precision (mAP) For multi task learning with pre trained weights (MultiCheXNet) with the dataset RSNA + SIIM-ACR, it offered 73% accuracy and F1 score 0.73 for classification, dice score as 0.78 for segmentation, 0.19 as mean average precision (mAP)
Viral Pneumonia Screening on Chest X-rays Using Confidence-Aware Anomaly Detection [38]	The clinical X-VIRAL dataset and X-COVID dataset. The X-VIRAL dataset contains 5,977 viral pneumonia cases, 18,619 non-viral pneumonia cases, and 18,774 healthy controls (i.e., 5977 positive and 37,393 negative cases)	The authors have proposed a Confidence-aware anomaly detection (CAAD) model, which consists of a shared feature extractor, an anomaly detection module, and a confidence prediction module	The proposed CAAD model achieves an AUC of 83.61% on COVID-19 screening, which outperforms other AI-based methods
Improving multi-label chest X-ray disease diagnosis by exploiting disease and health labels dependencies [12]	ChestX-ray14 dataset: This dataset contains 112,120 frontal-view X-rays with 14 disease labels	The authors have proposed novel error functions, Multi-label Soft max Loss (MSML) and Correlation Loss with bilinear pooling encoder and discriminator for multi-scaled features for any deep learning model	Local ResNet-BL-Proposed with 0.8310% accuracy, Local-Fixed ResNet-BL-Proposed with 0.8356% accuracy and Global ResNet-BL-Proposed with 0.8388% accuracy

Table 2 Structure of dataset

	Train	Test	Validation	Total
Normal	1341	234	8	1583
Pneumonia	3875	390	8	4273
Total	5216	624	16	5856

the structure of the dataset. There are in total 5856 chest X-Rays out of which 1583 were from healthy or normal people and 4273 chest X-Ray scans were from persons who had Pneumonia. Our model was trained on a training set and evaluated on a test set.

3.2 Implementation environment and tools used:

The experiment was implemented in Python 3.6 and performed on Google Colaboratory (https://colab.research.google.com/), a Jupyter notebook environment. To speed up computation, we used a Tesla K80 GPU freely provided by Google. The Colaboratory environment provides free access to 12 GB of GDDR5 VRAM and 13 GB RAM. We got the Kaggle dataset with the API provided by Kaggle on Colab environment. For building deep Learning models we used Keras and Tensorflow library with versions 2.2.2 and 1.10.0 respectively.

3.3 Proposed architecture

We have proposed Deep quaternion residual network architecture on the image dataset. Figure 1 shows a zero level diagram of our proposed architecture. It started with acquisition of datasets followed by preprocessing which includes downsizing of images to 50×50 to avoid exhaustion of computation resources. Then these images were transformed into quaternion matrices so that they can input to our proposed quaternion residual network architecture. In our proposed model construction, we have used a custom quaternion convolution layer and fully connected layer where input and output will be quaternion values. In the model construction, we have taken 4 residual blocks; the structure of each residual block is shown in Fig. 10. Batch Normalization [37] has been used to speed up the training process with Exponential Linear Unit [6] as an activation function as an excellent substitute to Rectified Linear Unit (ReLU) activation function. It provides accurate results as it tends to zero faster and also it reduces the bias shift effects which are there in ReLU activation function.

why are we using Eulu

ReLU activation function is mathematically defined as given in Eq. 19 and shown in Fig. 8.

$$\begin{matrix} f(x) = x & if\ x > 0 \\ 0 & otherwise \end{matrix} \tag{19}$$

ELU activation function is mathematically defined as in Eq. 20 and shown in Fig. 9.

$$\begin{matrix} f(x) = x & if\ x > 0 \\ \propto (e^x - 1) & otherwise \end{matrix} \tag{20}$$

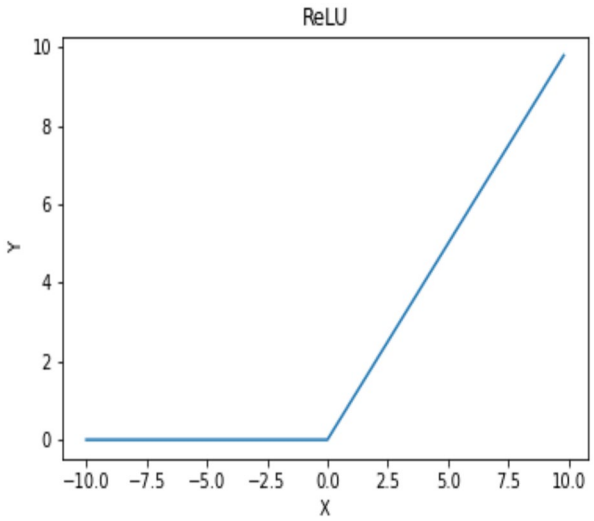


Fig. 8 Plot for ReLU Activation

3.3.1 Work Flow in the proposed architecture:

Figure 11 shows the flow of our experimentation. The basic modules are (a) Preparing the training set by preprocessing and converting the training image set into quaternions (b) Designing architecture and training on proposed architecture on the train set (c) preparing the test set by preprocessing and converting the test set into quaternions (d) Evaluating the trained model on the test set.

We acquired the data from Kaggle repository and resized all X-rays images to size 50×50 and then the dataset was split into a training set and a testing set. All images in training set were converted into quaternion values and then fed to our proposed architecture and then it is trained on normalized and preprocessed training dataset to produce trained models. Now the prediction is made by this trained model on a testing set to observe its classification accuracy and other performance metrics.

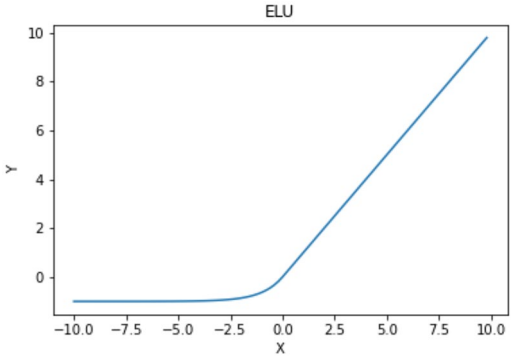


Fig. 9 Plot for ELU Activation

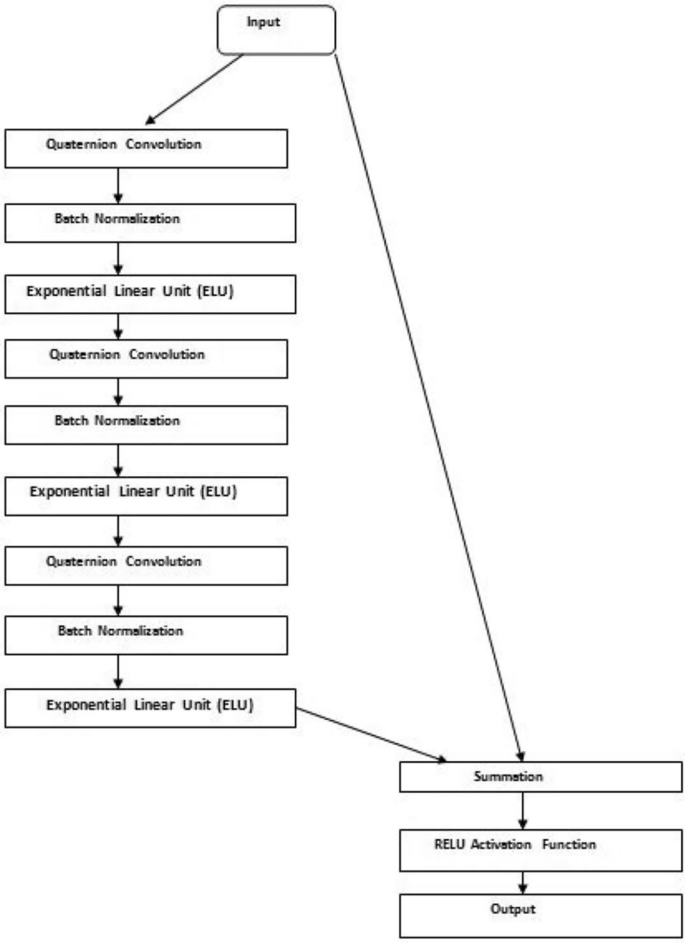


Fig. 10 Quaternion Residual Block Structure

3.3.2 Layer wise representation of our proposed model

In our proposed model as given in Fig. 12, main building blocks are Quaternion Convolution layer, Batch Normalization, ReLU, Max Pooling and Residual Blocks. In our paper, we experimented with 2, 3, 4, 5, 6 residual blocks and found that it gives better results in case of 4 residual blocks.

Quaternion Convolution Layer: this layer is the quaternion extension of the convolution layer so that it can take input as quaternion values and this layer enables to extract features of input in the form of a feature map. In our experimentation, the first quaternion convolution layer used 16 nodes and filter size was (1, 1) and this filter is shifted each time by a definite number of pixels which is also known as stride, in our implementation, stride was taken as (1,1).

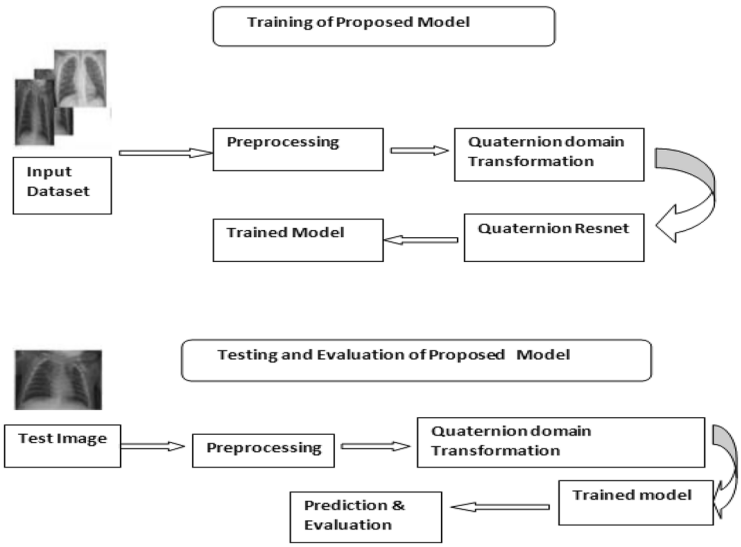


Fig. 11 Workflow in our proposed system

Batch Normalization: The purpose of this layer is to speed-up the training by normalizing selected layer's features by applying mean and standard deviation of inputs. It also reduces overfitting as it acts a regularizer same as dropout.

Rectified Linear Unit Activation Function(ReLU): It is the most commonly used activation function in most neural networks as it gets over vanishing gradient problem and the model will learn faster and deliver better performance.

Max Pooling: It is a down sampling technique and it gives the maximum value for each patch in the feature map and it reduces the dimensionality of the feature map.

Residual Blocks: this is shown in Fig. 10 and described in 2.2.2. In our proposed work there are 4 residual blocks and where stride is (1,1) and kernel is of size (1,1).

Hyper parameter in our experimentation is mentioned in Table 3

4 Results and discussions

This section presents the performance of our proposed model on testing set and visualization of evaluation indicators of the same. We have also evaluated the performance on the same dataset of other CNN architecture and presented the comparison in Table 5.

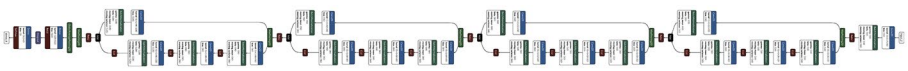


Fig. 12 Layered structure of our proposed architecture

Table 3 Hyper parameters for our proposed model

Parameters	Description
Stride	(1,1)
Loss function	Binary cross entropy
Decay	1e-6
Learning Rate	0.001
Optimizer	Adam
Image resize	50×50

4.1 Evaluation Indicators

To compare the performance of various deep learning architectures, most common evaluation indicators are classification accuracy, sensitivity, specificity, Area Under Curve(AUC) and confusion matrix.

Confusion Matrix: It is a table to show predictions made by the model in terms of TP, TN, FP and FN.

Table 4 shows comparison between predicted and actual labels. From this table we can evaluate other evaluation indicators.

Classification Accuracy: This indicator tells us about correct predictions made by our trained model on unseen new images from the test set. The mathematical formula of classification accuracy is as given in Eq. 21.

$$\text{Classification Accuracy} = \frac{\text{No.of correct prediction on test set}}{\text{size of test}} = \frac{(\text{TP+TN})}{(\text{TP+TN})+\text{FP+FN}} \tag{21}$$

True Positive (TP): Sick people correctly predicted as sick.

False Positive (FP): Normal people wrongly predicted as sick.

True negative(TN): Normal people correctly predicted as Normal.

False negative(FN): Sick people wrongly predicted as Normal.

Sensitivity: It is the proportion of positive samples that were classified as positive. Higher sensitivity indicates that the system can accurately predict the presence of disease and false negative cases will be very less. It is mathematically defined as in Eq. 22.

$$sensitivity = \frac{TP}{TP + FN} \tag{22}$$

Specificity: It shows the proportion of negative samples that were classified as negative. Higher specificity means that the system is making correct predictions about healthy people. It is defined as in Eq. 23.

$$specificity = \frac{TN}{TN + FN} \tag{23}$$

Table 4 Confusion Matrix

Prediction\Actual	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Table 5 Performance Evaluation of Proposed Model

Sr	epochs	Accuracy	F-score	Sensitivity	Specificity	AUC
1	20	63.67	0.5755	0.6690	0.9850	0.95
2	30	91.01	0.9222	0.9396	0.7391	0.94
3	40	91.40	0.9222	0.9417	0.7746	0.97
4	50	93.75	0.9640	0.9560	0.8311	0.97

Precision and Recall are defined as in Eq. 24

$$Precision = \frac{TP}{TP + FP} \; Recall = \frac{TP}{TP + FN} \tag{24}$$

F-Score is harmonic mean Precision (P) and Recall (R). It is defined as in Eq. 25.

$$F - score = \frac{2 * P * R}{(P + R)} \tag{25}$$

The Receiver Operator Characteristic (ROC) curve is graph between sensitivity and (1-specificity). It is used to analyze the tradeoff between sensitivity and specificity.

Area under Curve (AUC): Higher AUC means that the model can better distinguish between positive and negative classes.

We have repeated the experiments multiple times and observed the evaluation indicators of performance of the proposed model and variation of training and testing loss. In our experimentation, the ratio of training set and test set has been taken as 80 and 20. Table 5 shows accuracy and other performance metrics for 20, 30, 40 and 60 epochs respectively. For 40 epochs, our proposed architecture is proving an accuracy of 93.75% and F-score of 0.964. The accuracy plot and loss curve showing variation of accuracy and loss on training data and validation data is given in Figs. 13 and 14.

From Figs. 13 and 14, One can observe that there are fluctuation in accuracy up to 30 epochs and beyond 30 epochs, accuracy and loss have become stable. In order to get good performance we had to perform the experiment for 40 and 50 epochs.

Fig. 13 Accuracy Plot

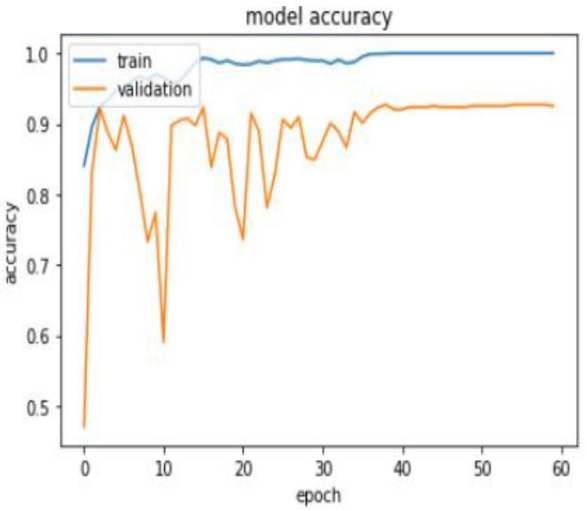
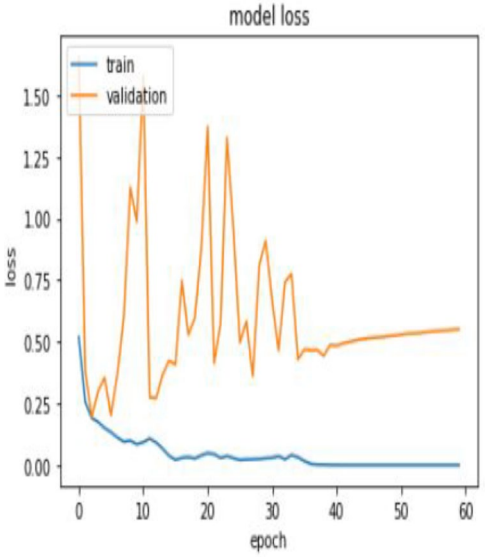


Fig. 14 Loss Curve



From the available ground truth, the plot for confusion matrix and AUC Curve is given in Figs. 15 and 16.

To perform comparison, we have also calculated the performance of other CNN architectures on the same dataset for 50 epochs and other hyper parameters same as mentioned in Table 3. Their performance is mentioned in Table 5. After the model was trained for 50

Fig. 15 Confusion Matrix

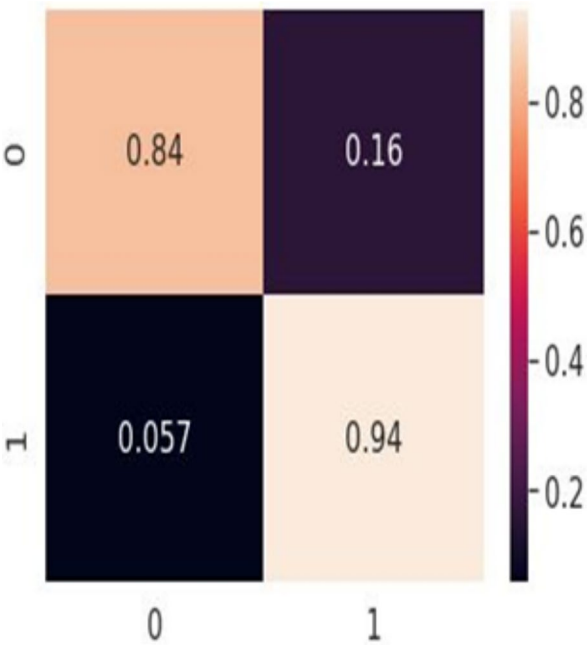
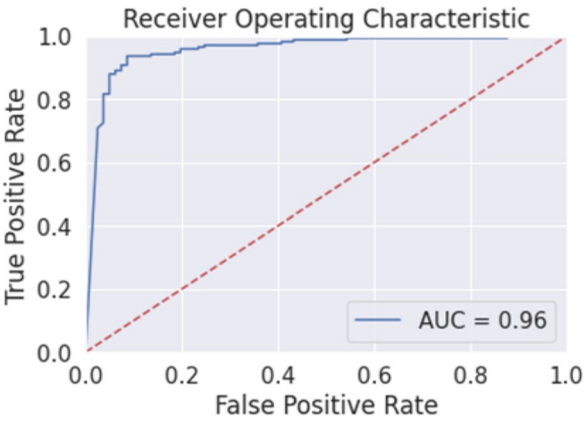


Fig. 16 ROC Curve



epochs, we observed its prediction on 16 random X-ray images. Here 0 indicates Normal and 1 indicates Pneumonia X-Ray Image.

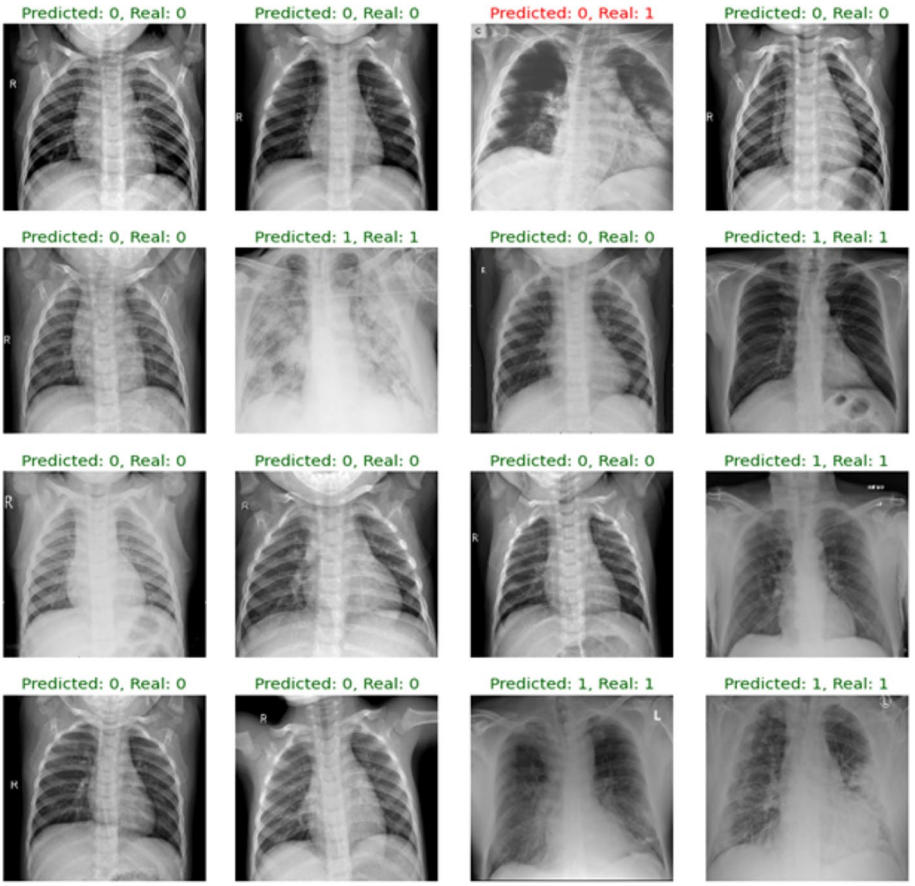


Fig. 17 Visualization of Prediction on test data

Table 6 Comparison with other deep learning architectures

Sr No	Architecture	Ref	Accuracy	F-score	# of trainable parameters	# of Non trainable parameters
1	VGG16	[23]	92.14	0.9234	50,178	14,714,688
2	VGG19	[28]	90.22	0.8999	50,178	20,024,384
3	ResNet50	[22]	82.37	0.8281	200,706	23,587,712
4	ResNet101	[29]	75.96	0.7593	200,706	42,658,176
5	ResNet152	[1]	87.179	0.8734	200,706	58,370,944
6	ResNet50V2	[35]	89.26	0.8937	200,706	23,564,800
7	ResNet101V2	[23]	92.62	0.925	200,706	42,626,560
8	ResNet152V2	[23]	92.94	0.9312	200,706	58,331,648
9	InceptionV3	[2]	89.42	0.8937	102,402	21,802,784
10	InceptionResNetV2	[18]	90.7	0.8989	200,706	58,331,648
11	DenseNet121	[8]	91.82	0.9171	100,354	7,037,504
12	DenseNet169	[16]	88.78	0.8874	163,074	12,642,880
13	DenseNet201	[34]	91.83	0.9171	188,162	18,321,984
14	NASNetLarge	[4]	88.14	0.8812	975,746	84,916,818
15	Quaternion Residual Network		93.75	0.9405	560,769	8,576

From Fig. 17, we can see that we got only one case of wrong prediction and for rest we got correct prediction.

From Tables 5 and 6 and Fig. 18, we see that our proposed model has delivered classification accuracy of 93.75%, sensitivity as 0.95, specificity as 0.83 and AUC as 0.96 which are higher than that of any real valued residual net. Based on that we can say that our proposed model can predict presence and absence of Pneumonia with high accuracy and it can very well distinguish between two classes of Normal and Pneumonia.

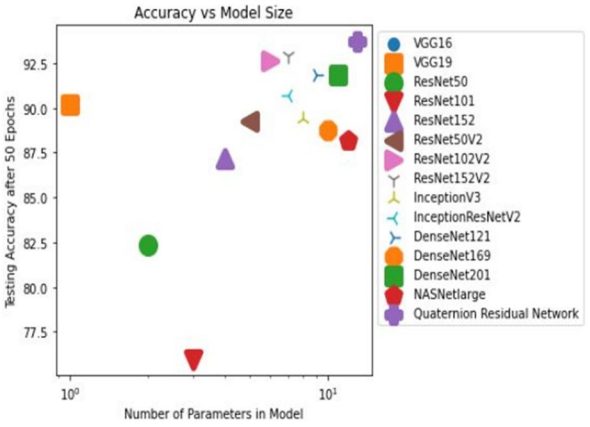


Fig. 18 Comparison of Classification Accuracy on various Architectures

5 Conclusion

In this paper, we have proposed a quaternion residual network for classification of Pneumonia from Chest X-ray. QCNN has already given promising results for speech classification, understanding of natural languages, image segmentation. We have extended the residual network to its quaternion domain and applied it for classification of chest X-ray images into Pneumonia or normal images and achieved an accuracy of 93.75 and F-score of 0.94 which are higher than that of real valued residual network. We have also evaluated the performance of other deep learning models which are usable as pre-trained weights by applying transfer learning on same dataset and found out that our proposed has outperformed these architectures that too, using less computation power and less number of epochs since these pre-trained deep learning models have already been trained on very huge dataset on a high end computer system. During the experimentation, we have faced the challenges of class imbalance and dataset size. Resolving these challenges by using a huge labeled balanced dataset, the performance can further be improved and made more reliable. The way we have extended the residual network into quaternion domain and other faster architecture can also be similarly customized into quaternion and Octonion domain which can solve other classification problems with improved performance which is left as future scope for this work.

References:

1. Abubakar A, Ajuji M, Usman Yahya I (2020) Comparison of Deep Transfer Learning Techniques in Human Skin Burns Discrimination. Appl Syst Innov. https://doi.org/10.3390/asi3020020
2. Agarwal R, Diaz O, Lladó X et al (2019) Automatic mass detection in mammograms using deep convolutional neural networks. J Med Imaging. https://doi.org/10.1117/1.jmi.6.3.031409
3. Aizenberg I (2011) Complex-valued neural networks with multi-valued neurons. Stud Comput Intell. https://doi.org/10.1007/978-3-642-20353-4_1
4. Albahli S, Albattah W (2020) Detection of coronavirus disease from X-ray images using deep learning and transfer learning algorithms. J Xray Sci Technol. https://doi.org/10.3233/XST-200720
5. Babu RV, Suresh S (2011) Fully complex-valued ELM classifiers for human action recognition. Proc Int Jt Conf Neural Networks 2803–28. https://doi.org/10.1109/IJCNN.2011.6033588
6. Clevert DA, Unterthiner T, Hochreiter S (2016) Fast and accurate deep network learning by exponential linear units (ELUs). In: 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings
7. Comminiello D, Lella M, Scardapane S, Uncini A (2019) Quaternion Convolutional Neural Networks for Detection and Localization of 3D Sound Events. In: ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings
8. Ezzat D, Ella HA (2020) GSA-DenseNet121-COVID-19: a Hybrid Deep Learning Architecture for the Diagnosis of COVID-19 Disease based on Gravitational Search Optimization Algorithm. ArxivOrg
9. Farag AT, El-Wahab ARA, Nada M et al (2020) MulticheXnet: A multi-task learning deep network for pneumonia-like diseases diagnosis from X-ray scans. arXiv
10. Gabruseva T, Poplavskiy D, Kalinin A (2020) Deep learning for automatic pneumonia detection. IEEE Comput Soc Conf Comput Vis Pattern Recognit Work 2020-June:1436–1443. https://doi.org/10.1109/CVPRW50498.2020.00183
11. Gaudet CJ, Maida AS (2018) Deep Quaternion Networks. In 2018 International Joint Conference on Neural Networks. IEEE, Rio de Janeiro, pp 1–8
12. Ge Z, Mahapatra D, Chang X et al (2020) Improving multi-label chest X-ray disease diagnosis by exploiting disease and health labels dependencies. Multimed Tools Appl 79:14889–14902. https://doi.org/10.1007/s11042-019-08260-2
13. Hardt M, Recht B, Singer Y (2016) Train faster, generalize better: Stability of stochastic gradient descent. In: 33rd International Conference on Machine Learning, ICML 2016

14. He K, Zhang X, Ren S, Sun J (2016) Deep Residual Learning for Image Recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Las Vegas, pp 770–778
15. Hirose A (2009) Complex-valued neural networks : The merits and their origins. 1237–1244
16. Huang GL, He J, Xu Z, Huang G (2020) A combination model based on transfer learning for waste classification. Concurr Comput. https://doi.org/10.1002/cpe.5751
17. Isokawa T, Matsui N, Nishimura H (2011) Quaternionic Neural Networks Complex-Valued Neural Networks 411–439. https://doi.org/10.4018/978-1-60566-214-5.ch016
18. Kensert A, Harrison P, Spjuth O (2018) Transfer learning with deep convolutional neural networks for classifying cellular morphological changes. bioRxiv. https://doi.org/10.1101/345728
19. Kermany D, Zhang K, Goldbaum M (2018) Large Dataset of Labeled Optical Coherence Tomography (OCT) and Chest X-Ray Imagesn. In: Mendeley Data, v3
20. Kumar S, Tripathi BK (2017) Machine learning with resilient propagation in quaternionic domain. Int J Intell Eng Syst 10:205–216. https://doi.org/10.22266/ijies2017.0831.22
21. Lan R, Zhou Y (2016) Quaternion-Michelson Descriptor for Color Image Classification. IEEE Trans Image Process. https://doi.org/10.1109/TIP.2016.2605922
22. Long J, Shelhamer E, Darrell T (2015) Fully Convolutional Networks for Semantic Segmentation. IEEE Trans Pattern Anal Mach Intell 39:640–651. https://doi.org/10.1109/TPAMI.2016.2572683
23. Muneeb ul Hassan (2018) VGG16 - Convolutional Network for Classification and Detection. In: Neurohive
24. Parcollet T, Zhang Y, Morchid M et al (2018) Quaternion Convolutional Neural Networks for End-to-End Automatic Speech Recognition
25. Parcollet T, Morchid M, Linares G (2019) Quaternion Convolutional Neural Networks for Heterogeneous Image Processing. ICASSP, IEEE Int Conf Acoust Speech Signal Process - Proc 2019-May:8514–8518. https://doi.org/10.1109/ICASSP.2019.8682495
26. Parcollet T, Ravanelli M, Morchid M et al (2019) Quaternion recurrent neural networks. 7th Int Conf Learn Represent ICLR 2019 1–19
27. Rajpurkar P, Irvin J, Zhu K et al (2017) CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning. arXiv 3–9
28. Shaha M, Pawar M (2018) Transfer Learning for Image Classification. In: Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, ICECA 2018
29. Shallu MR (2018) Breast cancer histology images classification: Training from scratch or transfer learning? ICT Express. https://doi.org/10.1016/j.icte.2018.10.007
30. Simonyan K, Zisserman A (2015) VGGNet. 3rd Int Conf Learn Represent ICLR 2015 - Conf Track Proc
31. Szegedy C, Toshev A, Erhan D (2013) Deep Neural Networks for object detection. In: Advances in Neural Information Processing Systems
32. Tripathi BK (2015) High Dimensional Neurocomputing. Stud. Comput Intell 571:79–103
33. Tripathi BK, Kalra PK (2010) High dimensional neural networks and applications. In: Pratihar DKJL (ed) Intelligent Autonomous Systems. Springer, Berlin Heidelberg, Kanpur, pp 215–233
34. Turkoglu M, Hanbay D, Sengur A (2019) Multi-model LSTM-based convolutional neural networks for detection of apple diseases and pests. J Ambient Intell Humaniz Comput. https://doi.org/10.1007/s12652-019-01591-w
35. Van Hieu N, Hien NLH (2020) Recognition of plant species using deep convolutional feature extraction. Int J Emerg Technol
36. Vince J (2008) Quaternion Algebra. In: Geometric Algebra for Computer Graphics
37. Wu Y, He K (2018) Group normalization. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)
38. Zhang J, Xie Y, Pang G et al (2020) Viral Pneumonia Screening on Chest X-rays Using Confidence-Aware Anomaly Detection IEEE Trans Med Imaging 1–1. https://doi.org/10.1109/tmi.2020.3040950
39. Zhu X, Xu Y, Xu H, Chen C (2018) Quaternion convolutional neural networks. Proceedings of the European Conference on Computer Vision (ECCV) 2018:631–647
40. Zhu X, Xu Y, Xu H, Chen C (2018) Quaternion convolutional neural networks. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) LNCS11212:645–661. https://doi.org/10.1007/978-3-030-01237-3_39