Squeeze and Hypercomplex Networks on Leaf Disease Detection

Nazmul Shahadat*, Anh Nguyen[1], Ritika Lama[1]

Truman State University, Kirksville, Missouri, USA

Abstract. Detecting agricultural leaf disease is critical for crop yield and quality, where deep attention models offer promising solutions over traditional methods. This paper introduces a novel approach utilizing Squeeze-and-Hypercomplex networks (SHNets) to detect and classify leaf diseases. The existing Squeeze-and-Excitation network (SENet) enhances feature representation through channel-wise (all channels) feature re-calibration. Unlike this, Parameterized hypercomplex multiplication (PHM) based hypercomplex dense layer is used to calculate cross-channel correlations across channels. This enhances the network's representational capacity by adaptively recalibrating cross-channel feature maps and sharing weights among channels. We introduce a novel hypercomplex dense layer to inherit hypercomplex advantages in SE-based attention networks. Moreover, using hypercomplex algebra in network design enables more expressive modeling of inter-channel dependencies, capturing complex patterns in leaf imagery. Our proposed SHNet architecture was trained and evaluated on diverse leaf disease datasets, including disease categories and healthy samples. The experimental results on benchmark datasets unequivocally demonstrate the superiority of our proposed SHNet over the state-of-the-art SENet methods in terms of accuracy and computational complexity. This makes SHNet a highly suitable solution for real-time applications in precision agriculture, where the timely detection and classification of leaf diseases can significantly impact crop yield and quality.

Keywords: SHNet · Squeeze and Hypercomplex Networks · Squeeze-and-Excitation networks · Attention networks · Leaf Disease detection.

1 Introduction

Agriculture is crucial for global economic growth and human survival, as all life depends on food. According to the United States Department of Agriculture (USDA), agriculture-related industries contributed roughly $1.530 trillion to the U.S. GDP, 4% of the global GDP, and more than 25% for developing countries' GDP in 2023. Every country prioritizes investing in agricultural innovation, policies, and infrastructure. Due to rapid population growth, urbanization, and

* Corresponding Author

[1] Authors contributed equally

climate change, fertile farmland is depleting, hindering crop growth. Innovative methods are needed to grow crops in less fertile soils. Researchers have developed various technologies to enhance crop production.

Due to poor soil minerals and unfavorable climates, crops can develop various diseases, such as spots, dead tissue, discoloration, wilting, stunted growth, and damage [21]. About 85% of plant diseases are caused by fungi and Non-infectious factors, such as nutrient deficiencies or temperature extremes [9]. According to the USDA, plant diseases cost the global economy around $220 billion annually and can cause a loss of 20–40% of global crop production. Plant pathogens and pests are responsible for up to 40% of maize, potato, rice, soybean, and wheat crop yield losses worldwide [25]. Therefore, farmers often use chemical fertilizers haphazardly, highlighting the importance of disease detection in agriculture.

Plants are susceptible to numerous diseases that can significantly impact crop health. Therefore, the diagnosis of plant diseases is crucial to mitigate these risks. Many Artificial Intelligence(AI) and Machine learning(ML) techniques, especially various convolutional neural networks (CNNs) (VGG [31], MobileNet [4, 20], ResNet [12, 13]), have shown exceptional abilities in diagnosing plant leaf diseases. No one has used an attention mechanism to analyze the diseases of these crops. Several attention-based networks have been introduced, including SENet [8], known for its parameter-efficient architecture. Moreover, the SE blocks recalibrate feature responses to focus on the most informative features and capture dependencies across channels. However, it uses two fully connected (FC) layers, which consumes high costs.

This research introduces a novel parameterized hypercomplex multiplication (PHM)-based FC layer that inherits all properties of SENet and introduces hypercomplex properties that provide better representational feature maps. The PHM layers require fewer computational resources, allow more efficient representations, and construct efficient and robust network using hypercomplex algebra. We analyze some major crop diseases, including rice, corn, wheat, and some others, using this better representational attention mechanism called the Squeeze-and-Hypercomplex network (SHNet). The effectiveness of our SHNet model is demonstrated experimentally on four crop disease detection datasets. Our assessments are based on parameter counts, FLOPs, and testing accuracy.

2 Literature Reviews

2.1 Rice Leaf Diseases Detection

In 2020, Matin et al. used the AlexNet technique to detect rice leaf diseases and demonstrated more than 99% accuracy [15]. In 2021, the model provided by Kathiresan et al., which was tested on a GAN-augmented dataset, achieved an average accuracy of 98.79% [11]. Bari et al. proposed a Faster R-CNN in diagnosing the three rice leaf diseases with accuracy rates of 98.09%, 98.85%, and 99.17%, respectively [3]. Mohapatra et al. used pre-trained InceptionV3 and ResNet152 and achieved a higher accuracy of 97.47% [16]. Yang et al. introduced the DHLC-FPN for the IDADP dataset and achieved 97.44% accuracy [34].

2.2 Wheat Leaf Diseases Detection

Rathore et al. developed a hybrid model called WheCNet, which achieved a validation accuracy of 98% [22]. Kumari et al. analyzed the ResNet model on the wheat leaf disease detection dataset, which attained the highest classification accuracy of 98% [14]. Saraswat et al. proposed a methodology for accurately detecting eight different types of wheat leaf disease [24] and revealed that the model beats all other models, with a classification accuracy of 98.08%.

2.3 Corn Leaf Diseases Detection

VGG16, a CNN version, is used to categorize infected and healthy leaves in a study by Subramanian et al. [31] and recorded an accuracy of 97%. Olayiwola et al. proposed a CNN-based model to identify four corn diseases with a 98.56% accuracy rate [17]. Kumar Sharma et al. utilized ResNext101, ResNext50, and Inception V3, achieving average accuracy levels of 91.59%, 88.43%, and 78.5%, respectively [13]. Yeswanth et al. proposed the ASFESRN model to analyze the PlantVillage Corn Leaf disease dataset and achieved accuracies of 99.7402%, 98.4805%, and 98.961% for X2, X4, X6 image scaling factors, respectively [35].

2.4 New Plant Leaf Diseases Data

Working on an open dataset containing 15200 photos of crop leaves, a ResNet34 was trained by Kumar et al. in 2020, which achieved 99.40% accuracy [12]. Pandian et al. presented a deep CNN with an average testing accuracy of 98.1% [18]. A new plant leaf dataset of 10,851 images of 44 different diseases was tested using a CNN-based attention model achieving an accuracy of 97.33% [36]. An article in 2023 by Binnar et al. detects leaf disease using three deep-learning models: AlexNet, MobileNet, and Inception-v3. They concluded that the MobileNet model is an excellent fit for the plant diseases dataset, with an accuracy of 97.52% [4]. Alqahtani et al. proposed the PlantRefineDet, which tested the PlantVillage data and achieved an accuracy of 99.99% [1].

3 Background Works

3.1 Residual 1D Convolutional Networks

Shahadat and Maida proposed a residual 1D CNN (RCN) to replace any 2D spatial CNN layer in a network to reduce the cost further. They replaced any block's 2D CNN with two 1D DSC layer. To use 1D DSC in 2D input. They separated the inputs into height and width axes for a size of $h \times w$. Each 1D CNN layer is applied to each input axis. The 1D DSC operation is defined in [27]. The n_{th} channel of trainable weight W is applied to the n_{th} of input X to get the n_{th} channel of the output feature map C_o. The computational cost of 1D DSC in RCN is, $Cost_{Conv1D} = h \cdot d_{out} \cdot k$, where d_{out} is the output channel counts. As the RCN block has two layers of 1D CNN, the total cost is twice the original cost. This type of network is used to avoid the vanishing gradient problem.

3.2 Quaternion Convolution Networks

[2] was the first to employ the quaternion number system in a neural network. Quaternion numbers are made up of one real and three imaginary components, expressed as, $Q = r + ix + jy + kz$, where r, x, y, and z are real numbers and i, j, and k are the imaginary vectors. The Quaternion number system expands the 2D complex number system to four dimensions. A quaternion vector $Q_V = r+ix+jy+kz$ is convolved with a quaternion filter matrix $Q_F = R+iX+jY+kZ$, where R, X, Y, and Z are real-valued matrices and r, x, y, and z are real-valued vectors. This process is similar to complex convolution. The quaternion convolution (4D hypercomplex network) is defined as [29],

$$\begin{aligned} Q_F \circledast Q_V = &(Rr - Xx - Yy - Zz) \\ &+i(Rx + Xr + Yz - Zy) \\ &+j(Ry - Xz + Yr + Zx) \\ &+k(Rz + Xy - Yx + Zr). \end{aligned} \tag{1}$$

Every input vector is convolved with every kernel to expose cross-channel convolutional processes. The matrix representation of this quaternion CNN (QCNN) is defined in [29], where each kernel is shared among four convolutions, each with four input channels, which explains cross-channel weight sharing. Equation 1 shows 16 real-valued convolutions, but only four kernels are reused. Kernel reuse is how the channel weight sharing occurs [28].

3.3 Parameterized Hypercomplex Multiplication Layer

The main disadvantage of QCNN in neural networks is that it only exists in a relatively limited number of preset dimensions: 4D (Quaternions), 8D (Octonions), and 16D (Sedenions). The need to work within specific dimensionalities restricts the flexibility of neural network architectures using hypercomplex multiplication.

A Parameterized Hypercomplex Multiplication (PHM) is a generalized hypercomplex network proposed by [37]. PHM layers are more flexible than older techniques, learning multiplication rules directly from the data. The PHM FC layer transforms input x into output y, which is represented as, $y = PHM(x) = Hx + b$. Here, b represents a bias vector, and H represents a parameter matrix with dimensions $k \times d$, which is not a standard matrix but is constructed using a sum of Kronecker products. Our work uses four-dimensional PHM layer whose parameter matrix H is defined in Equation 2 [29].

Let the dimension of the PHM module be $D_{phm} = N$. The PHM operation requires that both d and k are divisible by N [29]. $\mathbf{H}$ is the sum of Kronecker products of the parameter matrices, $\mathbf{A}_i \in \mathbb{R}^{N \times N}$ and $\mathbf{S}_i \in \mathbb{R}^{k/N \times d/N}$, where $i = 1 \dots N$, is defined as $\mathbf{H} = \sum_{i=1}^{N} A_i \otimes S_i$. Parameter reduction comes from reusing matrices A and S. The $\otimes$ is the Kronecker product. H is multiplied by the input in the dense layer. The learnable parameters for $N = 4$ are P_r, P_x, P_y, and P_z where $P \in \mathbb{R}^{1 \times 1}$. For A_i we use the hypercomplex matrix (4 dimensions), which is generated in a similar way to the vectormap convolution [28].

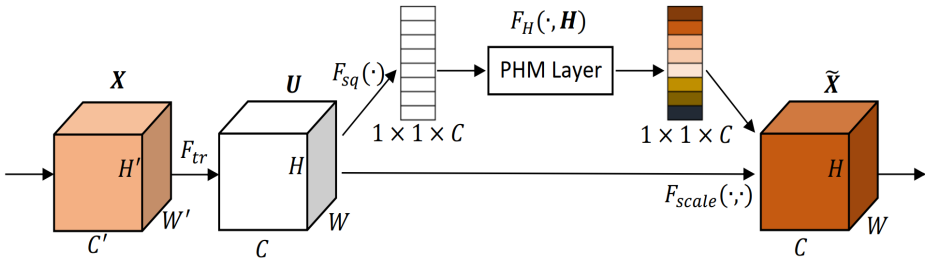


Fig. 1: Proposed Squeeze-and-Hypercomplex block.

$$H = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}}_{A_1} \otimes \underbrace{[P_r]}_{S_1} + \underbrace{\begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}}_{A_2} \otimes \underbrace{[P_x]}_{S_2} + \underbrace{\begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{A_3} \otimes \underbrace{[P_y]}_{S_3}$$
$$+ \underbrace{\begin{bmatrix} 0 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}}_{A_4} \otimes \underbrace{[P_z]}_{S_4} = \begin{bmatrix} P_r & -P_x & -P_y & -P_z \\ P_x & P_r & P_z & -P_y \\ P_y & -P_z & P_r & P_x \\ P_z & P_y & -P_x & P_r \end{bmatrix} \tag{2}$$

3.4 Squeeze-and-Excitation Network

Squeeze-and-Excitation Network (SENet) are designed in a way to enhance CNNs performance by capturing channel dependencies with fewer computational costs. SENet introduces additional parameters to each channel in a convolutional block, enabling the network to adjust the importance of each feature map and weigh each channel according to its significance [8]. SENet consists of two primary operations: Squeeze and Excitation. Squeeze operation compresses spatial dimensions while retaining channel information, using global pooling operations to condense spatial information into a channel descriptor. The excitation operation utilizes the channel descriptor generated during the squeeze operation and computes a set of channel-wise scaling factors [8]. These factors determine the importance of each channels.

If $V = [v_1, v_2, \ldots, v_c]$ is a set of learned filter channels, the outputs of F_{tr} is expressed as $U = [u_1, u_2, \ldots, u_c]$, where $u_c = v_c * X = \sum_{s=1}^{C'} v_c^s * x_s$. Here, $*$ represents convolution operation, and v_c^s is a 2D spatial kernel. The output is calculated by summing across all channels, embedding channel dependencies.

SENets have the capability to dynamically adjust the importance of each feature map, leading to improved performance in various tasks such as image classification and object detection. This capability makes SENets a powerful tool for enhancing CNNs without a great computational cost.

4 Proposed Squeeze-and-Hypercomplex Network

The local receptive fields of standard convolutional filters have limitations as they can only use contextual information within a limited portion of the input. Small receptive field sizes in the lowest network tiers exacerbate this problem. Conventional methods for mapping correlations across channels assume local receptive fields and may hinder the network's ability to discover dynamic, non-linear interactions between channels. In contrast, the SE block uses global information to uncover dynamic, non-linear connections across channels, boosting the network's representative strength and speeding up learning. The SE block is designed to enhance a network's representational power by explicitly modeling the interdependencies between the channels of its convolutional features. It achieves this by using global information to emphasize informative features and selectively suppress less useful ones. The SE block enables the network to perform feature recalibration, thereby enhancing CNN performance.

- The SE blocks recalibrate feature responses to focus on the most informative features and enhance representational capacity.
- The squeeze operation uses global average pooling to reduce each feature map to a single value, compressing the spatial dimensions of each channel into a single value.
- The excitation operation uses two fully connected (FC) layers to generate channel-wise weights. The first FC layer reduces the dimensionality and captures dependencies across channels, while the second FC layer restores the original channel dimensions.
- The excitation operation result undergoes a sigmoid activation function.
- The recalibration step emphasizes important channels and diminishes less important ones, leading to improved feature representation.
- SE blocks can seamlessly enhance existing CNN architectures to learn useful features without significantly increasing computational costs.
- SE blocks add relatively few parameters and computations compared to the overall network, making them efficient in terms of resource usage.

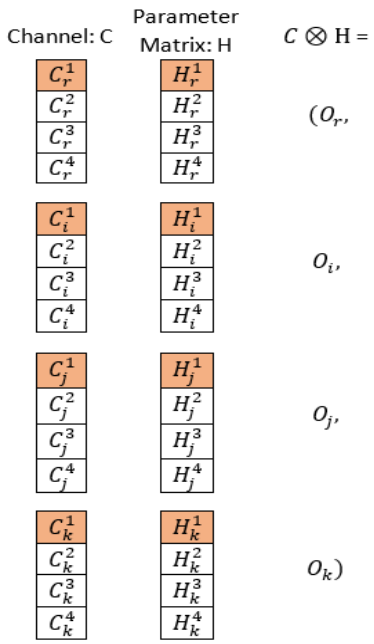


Fig. 2: Single valued quaternion channels and hypercomplex parameter matrix are represented using four channels of real values.

However, the SE block utilizes two FC layers, which is expensive. Without an explicit mechanism, the SE block diminishes the channel counts to reduce the number of parameters. Although the SE block re-calibrates input feature maps, it cannot establish cross-channel correlations. To address these limitations, we introduce a Squeeze-and-Hypercomplex network (SHNet).

Like the Squeeze-and-Excitation network (SENet), We have two stages of network operations: Squeeze and Hypercomplex. We use the squeeze operation to capture the global context of the input feature maps, applying global average pooling to reduce the feature map to a single scalar value. For a vision input with input height H, and width W, these are explained by the following equation [8],

$$z_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{ijc} \tag{3}$$

where x_{ijc} represents the value at spatial location (i, j) in channel c and z_c is the global descriptor for channel c.

The excitation layer, mainly two FC layers, was used to apply channel-wise dependencies, emphasize the important channels, and improve feature map representation using channel recalibration. However, we use a novel PHM layer, a generalized hypercomplex network (HCNN) designed for FC layer, to import the advantages provided by FC layer (Excitation) and the advantages (cross-channel correlation concept) provided by HCNNs. The HCNNs provide a better representational feature map [30], and reduce the trainable weights by a factor of $1/N$ for an N-dimensional HCNN [37]. Our proposed SHNet architecture is depicted in Figure 1.

To apply a hypercomplex network in the dense layer, we choose a PHM-based FC layer defined in Section 3.3, where H is the hypercomplex parameter matrix (HPM) calculated using the Equation 2. We rewrite this PHM FC layer as $y = PHM(x) = HC + b$, where, x, C, b, and H, $H \in \mathbb{R}^{N \times N}$, are the input image, the single-valued input channels, the bias, and the PHM, respectively. Although our proposed PHM is generalized, we are going to explain 4D HCNN dense layer or 4D PHM layer ($N = 4$). We use permutation to construct our parameter matrix using $H = H_r, H_i, H_j, H_k$. The parameter matrix for 4D PHM layer is used to calculate the output y using,

$$\begin{aligned}
\begin{bmatrix} \mathscr{R}(C * H) \\ \mathscr{I}(C * H) \\ \mathscr{J}(C * H) \\ \mathscr{K}(C * H) \end{bmatrix} &= \begin{cases} 1 & i = 1 \\ 1 & i = j \\ 1 & j = (i + i - 1) \ mod \ N \\ -1 & \text{else} \end{cases} \odot \begin{bmatrix} H^1 & H^2 & H^3 & H^4 \\ H^4 & H^1 & H^2 & H^3 \\ H^3 & H^4 & H^1 & H^2 \\ H^2 & H^3 & H^4 & H^1 \end{bmatrix} * \begin{bmatrix} C^1 \\ C^2 \\ C^3 \\ C^4 \end{bmatrix} \\
&= \begin{bmatrix} 1 & -1 & -1 & -1 \\ 1 & 1 & 1 & -1 \\ 1 & -1 & 1 & 1 \\ 1 & 1 & -1 & 1 \end{bmatrix} \odot \begin{bmatrix} H^1 & H^2 & H^3 & H^4 \\ H^4 & H^1 & H^2 & H^3 \\ H^3 & H^4 & H^1 & H^2 \\ H^2 & H^3 & H^4 & H^1 \end{bmatrix} * \begin{bmatrix} C^1 \\ C^2 \\ C^3 \\ C^4 \end{bmatrix}
\end{aligned} \tag{4}$$

where, $\odot$ is the element wise multiplication. So, the Hamiltonian product of two quaternion H (parameter matrix) and C (input channels) is defined as,

$$\begin{aligned} H \otimes C &= C^1H^1 + C^2H^2 + C^3H^3 + C^4H^4 \\ &= (C_r^1, C_i^1, C_j^1, C_k^1)(H_r^1, H_i^1, H_j^1, H_k^1) \\ &+ (C_r^2, C_i^2, C_j^2, C_k^2)(H_r^2, H_i^2, H_j^2, H_k^2) \\ &+ (C_r^3, C_i^3, C_j^3, C_k^3)(H_r^3, H_i^3, H_j^3, H_k^3) \\ &+ (C_r^4, C_i^4, C_j^4, C_k^4)(H_r^4, H_i^4, H_j^4, H_k^4) \end{aligned} \tag{5}$$

where $\otimes$ represents the Hamiltonian product [29], $H \otimes C$, and all of the other symbols in Equation 5 are quaternion numbers. The first line of Equation 5 (RHS) is also depicted in Figure 2, where the terms composing the multiplication operation shown in Equation 6 are highlighted in orange.

Furthermore, we expand the first term C^1H^1 on the right side of Equation 5 (shaded area for the input channel feature map and parameter matrix in Figure 2) by using the distributive property and grouping terms [29], defined as,

$$\begin{aligned} C^1H^1 =& (O_r, O_i, O_j, O_k) \\ =& (C_r^1, C_i^1, C_j^1, C_k^1)(H_r^1, H_i^1, H_j^1, H_k^1) \\ =& (C_r^1H_r^1 - C_i^1H_i^1 - C_j^1H_j^1 - C_k^1H_k^1, \\ & C_i^1H_r^1 + C_r^1H_i^1 + C_j^1H_k^1 - C_k^1H_j^1, \\ & C_j^1H_r^1 + C_r^1H_j^1 + C_k^1H_i^1 - C_i^1H_k^1, \\ & C_k^1H_r^1 + C_r^1H_k^1 + C_i^1H_j^1 - C_j^1H_i^1) \end{aligned} \tag{6}$$

where $\mathbf{re}(C^1H^1) = C_r^1H_r^1 - C_i^1H_i^1 - C_j^1H_j^1 - C_k^1H_k^1$, $\mathbf{i}(C^1H^1) = C_i^1H_r^1 + C_r^1H_i^1 + C_j^1H_k^1 - C_k^1H_j^1$, $\mathbf{j}(C^1H^1) = C_j^1H_r^1 + C_r^1H_j^1 + C_k^1H_i^1 - C_i^1H_k^1$, and $\mathbf{k}(C^1H^1) = C_k^1H_r^1 + C_r^1H_k^1 + C_i^1H_j^1 - C_j^1H_i^1$. It means convolving the real-valued input channel with the real-valued parameter matrix channel to obtain a real-valued scalar. Hence, the real part in Equations 5 equals O_r. Similarly, the other parts O_i, O_j, and O_k are defined as,

$$\begin{aligned} O_r =& \mathbf{r}(C^1H^1) + \mathbf{r}(C^2H^2) + \mathbf{r}(C^3H^3) + \mathbf{r}(C^4H^4) \\ O_i =& \mathbf{i}(C^1H^1) + \mathbf{i}(C^2H^2) + \mathbf{i}(C^3H^3) + \mathbf{i}(C^4H^4) \\ O_j =& \mathbf{j}(C_r^1H_r^1) + \mathbf{j}(C^2H^2) + \mathbf{j}(C^3H^3) + \mathbf{j}(C^4H^4) \\ O_k =& \mathbf{k}(C^1H^1) + \mathbf{k}(C^2H^2) + \mathbf{k}(C^3H^3) + \mathbf{k}(C^4H^4) \end{aligned} \tag{7}$$

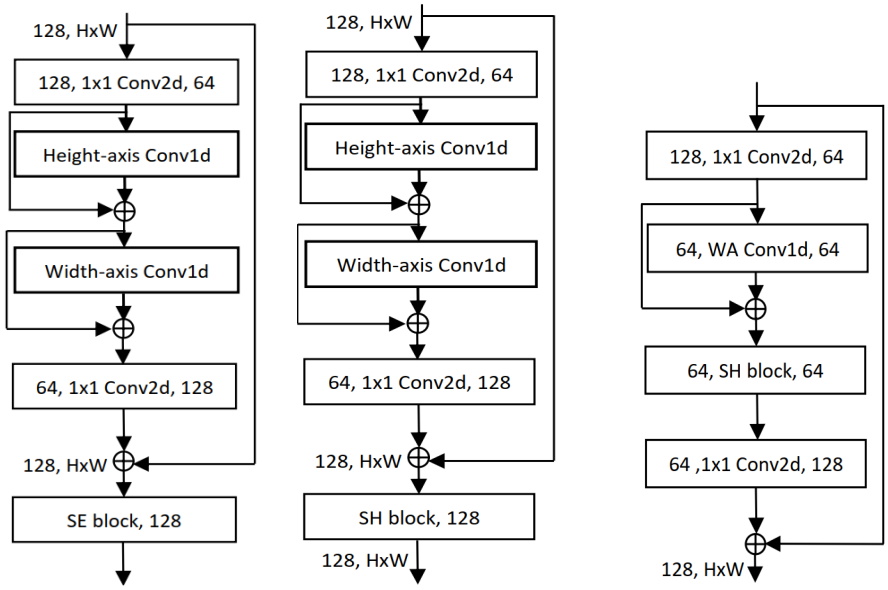
The real component (O_r) of the convolution value is defined as,

$$\begin{aligned} O_r &= C_r \otimes H_r - C_i \otimes H_i - C_j \otimes H_j - C_k \otimes H_k \\ &= C_r^1H_r^1 + C_r^2H_r^2 + C_r^3H_r^3 + C_r^4H_r^4 \\ &- C_i^1H_i^1 - C_i^2H_i^2 - C_i^3H_i^3 - C_i^4H_i^4 \\ &- C_j^1H_j^1 - C_j^2H_j^2 - C_j^3H_j^3 - C_j^4H_j^4 \\ &- C_k^1H_k^1 - C_k^2H_k^2 - C_k^3H_k^3 - C_k^4H_k^4 \end{aligned} \tag{8}$$

Similarly, we can define O_i, O_j, and O_k. Equation 6 can be reexpressed in matrix representation as,

$$\begin{bmatrix} \mathscr{R}(C^1 * H^1) \\ \mathscr{I}(C^1 * H^1) \\ \mathscr{J}(C^1 * H^1) \\ \mathscr{K}(C^1 * H^1) \end{bmatrix} = \begin{bmatrix} H_r^1 & -H_i^1 & -H_j^1 & -H_k^1 \\ H_i^1 & H_r^1 & -H_k^1 & H_j^1 \\ H_j^1 & H_k^1 & H_r^1 & -H_i^1 \\ H_k^1 & -H_j^1 & H_i^1 & H_r^1 \end{bmatrix} * \begin{bmatrix} C_r^1 \\ C_i^1 \\ C_j^1 \\ C_k^1 \end{bmatrix} \tag{9}$$

In Equation 9, each kernel channel is convolved with the corresponding image channel. Equation 6 and the graphical explanation in Figure 2 shows that each parameter of the trainable parameter matrix is used four times over the sixteen multiplications in Equations 5 and 9. Here, each input channel is convolved with all parameter matrix channels, which reveals cross-channel correlations. Parameter of the trainable parameter matrix reuse is how weight sharing occurs. [28,29] described weight sharing in the Hamilton product. The other three components of the sum in Equation 5 (i.e., C^2H^2, C^3H^3, C^4H^4) have the same structure as Equation 6, so the nature of the weight sharing is the same for all terms.



(a) SE-RCN (b) SH-RCN (c) Block with SHNet

Fig. 3: Block types. "SE" and "SH" stand for Squeeze-and-Excitation, and Squeeze-and-Hypercomplex, respectively. (a) SE layer applied after the RCN block, (b) SH layer applied after the RCN block, and (c) SHNet replaces SENet from 1D CNN with SENet found in [26].

We apply this new parameter-efficient hypercomplex FC layer to replace the FC layers in SENet and construct our novel parameter-efficient architecture called Squeeze-and-Hypercomplex network (SHNet). This SHNet architecture can provide better representational feature maps than the SENet. Also, SHNet reduces trainable parameters than the SENet in two ways: (1) the SHNet uses a hypercomplex FC layer than the two FC layers in SENet, and (2) the Hypercomplex FC layer consumes $1/N$ times fewer parameters than the real-valued FC layer. We apply our proposed SHNet block to replace the SENet from the existing parameter-efficient 1D CNN networks with SENet architecture, depicted in Figures 3a, 3b, and 3c. In these ways, any network can be constructed with our proposed SHNet architecture to boost the network performance and reduce the network's trainable parameters than the SENet.

5 EXPERIMENTAL RESULTS

5.1 Dataset Description

This paper experiments on the "Rice Leaf Disease" [23], "Wheat Leaf Disease" [10], "Corn Leaf Disease" [32], and "New Plant Leaf Disease" [33] kaggle datasets. Among these, the rice leaf disease dataset consists of $5,932$ images representing four distinct types of rice leaf diseases: Bacterial blight, Blast, Brown Spot, and Tungro [23]. The Wheat Leaf Disease Dataset includes five classes: $1,658$ images of healthy leaves, $1,256$ images of Brown Rust disease, 939 images of Loose Smut disease, 349 images of Septoria disease, and $1,395$ images of Yellow Rust disease [10]. Arun Pandian et al. contributed to the Corn Leaf Disease Dataset, a dataset for classifying corn or maize plant leaf diseases, in 2019. This dataset was based on two popular datasets: PlantVillage and PlantDoc [32]. We also constructed another corn dataset. We collected all corn disease and healthy images for four classes: 2052 images for gray spots, 2384 images for common rust, 2324 healthy images, and 2385 images for blight diseases from new plant disease dataset [33]. Moreover, the new Plant Diseases Dataset contains approximately 87K RGB images of healthy and diseased crop leaves, divided into 38 different categories [33]. We divided all of the datasets into two parts: 80% for the training set and 20% for the testing set.

Table 1: Cost comparisons of our proposed SHNet with 1D CNN (like the Figure 3c) and original SENet with 1D CNN architectures [26].

Architecture	Models	Params	FLOPs
23-1	SENet	0.33M	4.3M
	SHNet	0.3M	4.1M
23-2	SENet	1.4M	12.3M
	SHNet	1.1M	11.9M
44-1	SENet	0.58M	6.3M
	SHNet	0.54M	5.9M
44-2	SENet	2.4M	18.4M
	SHNet	2.06M	17.8M

5.2 Method

We applied our proposed SHNet to 1D CNN with SENet and constructed our block 1D CNN with SHNet. Like [26], which introduced 1D CNN with SENet, we also applied our proposed 1D CNN with SHNet block to SqueezeNext network architecture [7] and constructed our proposed network. We employed an identical training protocol as [26] for our proposed SqueezeNext (SqueezeNext block is replaced by our proposed 1D CNN with SHNet block) architectures to ensure a fair comparison. Like original work (1D CNN with SENet [26]), this work also utilizes the block multipliers "[6, 6, 8, 1]" and "[12, 12, 16, 2]", respectively, to construct 23-layer and 44-layer networks. We also analyzed these 23 and 44-layer architectures with widening factors 1 and 2, constructed 23-1, 23-2, 44-1, and 44-2 network architectures. Training was done on 64 batch sizes for all architectures. To investigate scalability, we trained the SGD (stochastic gradient descent) optimizer on the normalized input images. The images were normalized using the per-channel mean and standard deviation. All models were run using the linearly warmed-up learning for the first ten epochs from zero to 0.1 and then used cosine learning scheduling from epochs 11 to 150. The experiments were run on a workstation with an Intel(R) i9-13900K CPU, 128 GB memory, and NVIDIA RTX A6000 GPU (48GB).

Table 2: Performance evaluation on different leaf disease detection datasets using the 1D CNN with original SENet and our proposed SHNet architectures.

Dataset Type	Dataset Class	Models	Accuracy	
			SENet	SHNet
Wheat	5	23-1	96.93	98.68
		23-2	97.78	98.99
		44-1	97.27	98.87
		44-2	97.98	99.01
Rice	4	23-1	99.49	100
Corn	4	23-1	96.54	98.79
		23-2	97.31	98.89
		44-1	96.87	98.88
		44-2	97.37	98.91
Corn New Plant	4	23-1	98.91	99.29
		23-2	99.34	99.78
		44-1	99.01	99.35
		44-2	99.49	99.89
New Plant	38	23-1	99.41	99.94
		23-2	99.58	99.98
		44-1	99.47	99.96
		44-2	99.61	99.99

5.3 Result Analysis

This paper achieves a new height of performance accuracy in leaf disease detection. Our new proposed model yields impressive accuracy across multiple plant disease datasets. As previously described in section 5.1, we tested our model on wheat, rice, and corn plant datasets along with some other new plant disease detection datasets. We tested our proposed SHNet along with the original SENet model, and the experimental results are shown in Table 2. Even with fewer parameters and Flops (shown in Table 1), our novel model, when tested on the wheat leaf, was able to achieve 99.01% accuracy compared to 97.98% for SENet using 44-2 layers network. For all network architectures, our proposed SHNet performed better than the original SENet on wheat leaf data.

For the rice leaf disease dataset, our smallest model (23-1 network architecture) achieved 100% accuracy, whereas the original SENet performed 99.49. In the context of this data, it is notable that the original SENet architectures, with 23 and 44 layers and widening factors of 1 and 2, were unable to achieve the performance level demonstrated by the SHNet's 23-1 network architecture. Our SHNet was tested on corn data and corn new plant data and outperformed similarly to the wheat and rice data.

To analyze a dataset with more categories, we ran our model on a new plant disease dataset with 38 categories, where our SHNet beat the original SENet model for all network architectures; we tested our SHNet for 44 layer with a widening factor 2 achieved 99.99% performance.

Our SHNet model outperformed the SENet model across all datasets and models in this experimental analysis. In the case of cost comparison, Table 1 describes that our proposed SHNet consumes fewer parameters and FLOPs than the original SENet architecture in all the scenarios. 1D CNN with SENet and SHNet are used to evaluate these costs.

5.4 Comparison with the Literature

This section compares our new proposed model with other state-of-the-art performing models across four different datasets. SHNet outperformed other models like SVM, WheCNet, ResNet50, and DL Model by achieving an accuracy level of 99.01% in wheat leaf disease classification, shown in Table 3. Similarly, our model achieved a remarkable accuracy of 100% in the rice leaf dataset by beating the other models. Furthermore, we tested our model in the Corn leaf dataset. It achieved 99.89% accuracy, beating VGG16, ResNet, CNN, ResNext101, ASFESRN, and Ghost CNN. Finally, in the New Plant Disease dataset with 38 categories, our model once again achieved a remarkable accuracy of 99.99% compared to existing models like ResNet43, CNN, DCNN, MobileNet, and PlantRefineDet. Our SHNet model showed a similar level of accuracy as PlantRefineDet proposed by Algahtani et al. [1]. This is a remarkable level of performance achieved by our model, considering that PlantRefineDet is a computer-aided model, and our model is lightweight and mobile-embedded, which consumes only $2.06M$ parameters and $17.8M$ FLOPs. Our model's adaptability and outstanding performance, despite its lightweight design, confirms its potential in real-world scenarios.

5.5 Ablation Study

This section examines how different attention mechanisms (SE and ours SH) affect the RCN-based SqueezeNext model for detecting plant leaf diseases from the New Plant diseases dataset. We compared the RCN-based SqueezeNext 23-1 layer architectures regarding top-1 validation accuracy, parameters, and FLOPs as listed in Table 4. Without an attention model, RCN-SqueezeNext achieves an accuracy of 98.99% for $0.31M$ parameters and $4.08M$ FLOPs. Then, we utilized the SE attention layer in different network stages. The more SE attention layers are used, the better the network performs. Extending SE attention to stages 3

Table 3: State-of-the-art result studies on plant leaf disease detection datasets.

Category	Model	Architecture	Datasets	Accuracy
Wheat Leaf Diseases	El et al. [6]	SVM	Kaggle Data	98
	Rathore et al. [22]	WheCNet		98
	Kumari et al. [14]	ResNet50		98
	Saraswat et al. [24]	DL Model		98.08
	SH-RCN-SqueezeNext (Our)	44-2	Kaggle Data	**99.01**
Rice Leaf Diseases	Matin et al. [15]	AlexNet	Kaggle Data	99
	Pothen et al. [19]	SVM		94.6
	Kathiresan et al. [11]	RiceDenseNet	Open Sources	98.69
	Bari et al. [3]	Faster R-CNN	Open Sources	99.17
	Mohapatra et al. [16]	CNN	Open Sources	97.47
	Yang [34]	DHLC-FPN	IDADP	97.44
	SH-RCN-SqueezeNext (Our)	23-1	Kaggle Data	**100**
Corn Leaf Diseases	Subramanian et al. [31]	VGG16	Kaggle Data	97
	Olayiwola et al. [17]	CNN	Kaggle Data	98.56
	Kumar et al. [13]	ResNext101	Kaggle Data	91.59
	Yeswanth et al. [35]	ASFESRN	PlantVillage	99.74
	SH-RCN-SqueezeNext (Our)	44-1		**99.89**
New Plant Diseases	Kumar et al. [12]	ResNet34	Open Sources	99.4
	Deepalakshmi et al. [5]	CNN	Open Sources	94.5
	Pandian et al. [18]	DCNN	Open Sources	98.1
	Zamani et al. [36]	CNN	Open Sources	97.33
	Binnar et al. [4]	MobileNet	NPD	99.07
	Alqahtani et al. [1]	PlantRefineDet	PlantVillage	99.99
	SH-RCN-SqueezeNext (Our)	44-2	PlantVillage	**99.99**

Table 4: Analyze New Plant diseases dataset using attention mechanisms (without attention layer, SE attention, and our proposed SHNet attention) on different stages of the RCN-based SqueezeNext 23-1 layer architecture.

Models	Attention on Network stages				Params	FLOPs	Accuracy
	Stage 1	Stage 2	Stage 3	Stage 4			
RCN-SqueezeNext					0.31M	4.08M	98.99
SE-RCN-SqueezeNext				✓	0.316M	4.14M	99.07
SE-RCN-SqueezeNext			✓	✓	0.324M	4.23M	99.28
SE-RCN-SqueezeNext		✓	✓	✓	0.333M	4.31M	99.35
SE-RCN-SqueezeNext	✓	✓	✓	✓	0.335M	4.35M	99.41
SHNet-RCN-SqueezeNext				✓	0.42M	44.9M	99.77
SHNet-RCN-SqueezeNext			✓	✓	0.42M	45M	99.78
SHNet-RCN-SqueezeNext		✓	✓	✓	0.44M	45.2M	99.91
SHNet-RCN-SqueezeNext	✓	✓	✓	✓	0.31M	4.1M	99.94

and 4, the model increases its accuracy from 99.07 (performance when SE is only applied to stage 4) to 99.28% and reaches 99.35% when SE is applied in stages 2, 3, and 4. When SE attention is applied to all four stages, the model

achieved an accuracy of 99.41%, the highest among SE attention with $0.335M$ parameters and $4.35M$ FLOPs.

The integration of our proposed SH attention mechanism performs even better than that of the network with the SE layer. Applying SH attention at stage 4 alone leads to an accuracy of 99.77%, outperforming the performance of applying SE in all network stages. When SH attention is extended over stages 3 and 4, it yields an accuracy of 99.78%. Applying the SH attention mechanism on the last three stages and all four stages resulted in an overall accuracy of 99.91% and 99.94%. The results unequivocally show that the SH attention mechanism consistently enhances the model's performance compared to the SE mechanism, particularly when applied at multiple stages.

6 Conclusion

This study explores the efficacy of Squeeze-and-Hypercomplex Networks in detecting wheat, rice, corn, and new plant leaf diseases. This model exhibited superior performance in capturing complex patterns and relationships within the data. Using hypercomplex numbers, HCNNs can effectively model complex dependencies for improved disease detection accuracy, making them ideal for precise disease identification. The fusion of squeeze-and-excitation mechanisms with hypercomplex algebra provides a powerful and efficient framework for leaf disease detection, offering significant improvements in performance and scalability for related applications, and showing state-of-the-art results on some tested datasets. Our SHNet introduced cross-channel feature representation and feature recalibration, improving the model's performance.

This work's limitation is that the proposed model only tested four disease datasets. Due to machine limitations, we were unable to test some state-of-the-art datasets. Future work will focus on further optimizing these networks for other real-world applications and integrating this approach with some pretrained network architectures.

References

1. Alqahtani, Y., Nawaz, M., Nazir, T., Javed, A., Jeribi, F., Tahir, A.: An improved deep learning approach for localization and recognition of plant leaf diseases. Expert Systems with Applications **230**, 120717 (2023)
2. Arena, P., Fortuna, L., Occhipinti, L., Xibilia, M.G.: Neural networks for quaternion-valued function approximation. In: Proceedings of IEEE International Symposium on Circuits and Systems-ISCAS'94. vol. 6, pp. 307–310. IEEE (1994)
3. Bari, B.S., Islam, M.N., Rashid, M., Hasan, M.J., Razman, M.A.M., Musa, R.M., Ab Nasir, A.F., Majeed, A.P.A.: A real-time approach of diagnosing rice leaf disease using deep learning-based faster r-cnn framework. PeerJ Computer Science **7**, e432 (2021)
4. Binnar, V., Sharma, S.: Plant leaf diseases detection using deep learning algorithms. In: Machine Learning, Image Processing, Network Security and Data Sciences: Select Proceedings of 3rd International Conference on MIND 2021. pp. 217–228. Springer (2023)

5. Deepalakshmi, P., Lavanya, K., Srinivasu, P.N., et al.: Plant leaf disease detection using cnn algorithm. International Journal of Information System Modeling and Design (IJISMD) **12**(1), 1–21 (2021)
6. El-Sayed, R., Darwish, A., Hassanien, A.E.: Wheat leaf-disease detection using machine learning techniques for sustainable food quality. In: Artificial Intelligence: A Real Opportunity in the Food Industry, pp. 17–28. Springer (2022)
7. Gholami, A., Kwon, K., Wu, B., Tai, Z., Yue, X., Jin, P., Zhao, S., Keutzer, K.: Squeezenext: Hardware-aware neural network design. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 1638–1647 (2018)
8. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
9. Isleib, J.: Signs and symptoms of plant disease: Is it fungal, viral or bacterial? (Oct 2023), `https://www.canr.msu.edu/news/signs_and_symptoms_of_plant_disease_is_it_fungal_viral_or_bacterial`
10. jayaprakashpondy: wheat leaf disease. `https://www.kaggle.com/datasets/jayaprakashpondy/wheat-leaf-disease`, accessed: 2024-07-01
11. Kathiresan, G., Anirudh, M., Nagharjun, M., Karthik, R.: Disease detection in rice leaves using transfer learning techniques. In: Journal of Physics: Conference Series. vol. 1911, p. 012004. IOP Publishing (2021)
12. Kumar, V., Arora, H., Sisodia, J., et al.: Resnet-based approach for detection and classification of plant leaf diseases. In: 2020 international conference on electronics and sustainable communication systems (ICESC). pp. 495–502. IEEE (2020)
13. Kumar Sharma, N., Kalyani Immadisetty, B., Govina, A., Chandra Reddy, R., Choubey, P.: Corn leaf disease detection using resnext50, resnext101, and inception v3 deep neural networks. In: Machine Vision and Augmented Intelligence: Select Proceedings of MAI 2022, pp. 303–313. Springer (2023)
14. Kumari, N., Saini, B.: Fully automatic wheat disease detection system by using different cnn models. In: Sentiment Analysis and Deep Learning: Proceedings of ICSADL 2022, pp. 351–365. Springer (2023)
15. Matin, M.M.H., Khatun, A., Moazzam, M.G., Uddin, M.S.: An efficient disease detection technique of rice leaf using alexnet. Journal of Computer and Communications **8**(12), 49–57 (2020)
16. Mohapatra, S., Marandi, C., Sahoo, A., Mohanty, S., Tudu, K.: Rice leaf disease detection and classification using a deep neural network. In: International Conference on Computing, Communication and Learning. pp. 231–243. Springer (2022)
17. Olayiwola, J.O., Adejoju, J.A.: Maize (corn) leaf disease detection system using convolutional neural network (cnn). In: International Conference on Computational Science and Its Applications. pp. 309–321. Springer (2023)
18. Pandian, J.A., Kanchanadevi, K., Kumar, V.D., Jasińska, E., Goňo, R., Leonowicz, Z., Jasiński, M.: A five convolutional layer deep convolutional neural network for plant leaf disease detection. Electronics **11**(8), 1266 (2022)
19. Pothen, M.E., Pai, M.L.: Detection of rice leaf diseases using image processing. In: 2020 fourth international conference on computing methodologies and communication (ICCMC). pp. 424–430. IEEE (2020)
20. Ramadan, S.T.Y., Sakib, T., Haque, M.M.U., Sharmin, N., Rahman, M.M.: Generative adversarial network-based augmented rice leaf disease detection using deep learning. In: 2022 25th International Conference on Computer and Information Technology (ICCIT). pp. 976–981. IEEE (2022)
21. Randaci, A.: Common plant diseases (Nov 2021), `https://earthsally.com/disease-control/common-plant-diseases.html`

22. Rathore, N.P.S., Prasad, L.: Hybrid deep learning model to detect uncertain diseases in wheat leaves. Journal of Uncertain Systems **15**(03), 2241004 (2022)
23. Sankalana, N.: rice leaf disease image. `https://www.kaggle.com/datasets/nirmalsankalana/rice-leaf-disease-image?resource=download`, accessed: 2024-07-01
24. Saraswat, S., Batra, S., Neog, P.P., Sharma, E.L., Kumar, P.P., Pandey, A.K.: An efficient diagnostic approach for multi-class classification of wheat leaf disease using deep transfer and ensemble learning. In: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT). pp. 544–551. IEEE (2024)
25. Savary, S., Willocquet, L., Pethybridge, S.J., Esker, P., McRoberts, N., Nelson, A.: The global burden of pathogens and pests on major food crops. Nature ecology & evolution **3**(3), 430–439 (2019)
26. Shahadat, N.: Mobile-based deep convolutional networks for malaria parasites detection from blood cell images. In: 2023 26th International Conference on Computer and Information Technology (ICCIT). pp. 1–6. IEEE (2023)
27. Shahadat, N., Maida, A.S.: Deep residual axial networks. arXiv preprint arXiv:2301.04631 (2023)
28. Shahadat, N., Maida, A.S.: Enhancing resnet image classification performance by using parameterized hypercomplex multiplication. arXiv preprint arXiv:2301.04623 (2023)
29. Shahadat, N., Maida, A.S.: Cross channel weight sharing for image classification. Image and Vision Computing **141**, 104872 (2024)
30. Shahadat, N., Maida, A.S.: Improving axial-attention network via cross-channel weight sharing. In: The International FLAIRS Conference Proceedings. vol. 37 (2024)
31. Subramanian, M., Lv, N.P., VE, S.: Hyperparameter optimization for transfer learning of vgg16 for disease identification in corn leaves using bayesian optimization. Big Data **10**(3), 215–229 (2022)
32. Unknown6874: corn leaf disease dataset. `https://www.kaggle.com/datasets/unknown6874/corn-leaf-disease-dataset`, accessed: 2024-07-01
33. Vipoooool: new plant diseases dataset. `https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset`, accessed: 2024-07-01
34. Yang, H., Deng, X., Shen, H., Lei, Q., Zhang, S., Liu, N.: Disease detection and identification of rice leaf based on improved detection transformer. Agriculture **13**(7), 1361 (2023)
35. Yeswanth, P., Deivalakshmi, S.: Asfesrn: bridging the gap in real-time corn leaf disease detection with image super-resolution. Multimedia Systems **30**(4), 175 (2024)
36. Zamani, A.S., Anand, L., Rane, K.P., Prabhu, P., Buttar, A.M., Pallathadka, H., Raghuvanshi, A., Dugbakie, B.N.: [retracted] performance of machine learning and image processing in plant leaf disease detection. Journal of Food Quality **2022**(1), 1598796 (2022)
37. Zhang, A., Tay, Y., Zhang, S., Chan, A., Luu, A.T., Hui, S.C., Fu, J.: Beyond fully-connected layers with quaternions: Parameterization of hypercomplex multiplications with $1/n$ parameters. arXiv preprint arXiv:2102.08597 (2021)