Deep Separable Hypercomplex Networks

Nazmul Shahadat, Anthony S. Maida
University of Louisiana at Lafayette
Lafayette, Louisiana, USA

Abstract

Deep hypercomplex-inspired convolutional neural networks (CNNs) have recently enhanced feature extraction for image classification by allowing weight sharing across input channels. This makes it possible to improve the representation acquisition abilities of the networks. Hypercomplex-inspired networks, however, still incur higher computational costs than standard CNNs. This paper reduces this cost by decomposing a quaternion 2D convolutional module into two consecutive separable vectormap modules. In addition, we use 4 and 5D parameterized hypercomplex multiplication-based fully connected layers. Incorporating both yields our proposed hypercomplex CNN, a novel architecture that can be assembled to construct deep separable hypercomplex networks (SHNNs) for image classification. We conduct experiments on CIFAR, SVHN, and Tiny ImageNet datasets and achieve better performance using fewer trainable parameters and FLOPS. Our proposed model achieves almost 2% higher performance for CIFAR and SVHN datasets and more than 3% for the ImageNet-Tiny dataset and takes 84%, 35%, and 51% fewer parameters than the ResNets, quaternion, and vectormap networks, respectively. Also, we achieve state-of-the-art performance on CIFAR benchmarks in hypercomplex space.

Introduction

Convolutional neural networks (CNNs) and hypercomplex CNNs (HCNNs) for image classification form a hierarchical design where different layers extract different levels of feature representation. CNNs have shown significant success in recent decades (Buyssens, Elmoataz, and Lézoray, 2012; Javanmardi et al., 2021). In vision tasks, these CNN-based feature extraction designs can be improved in regard to working with multi-dimensional data. To enhance the CNNs ability, HCNNs have been used which treat the multi-dimensional data as a cohesive entity by applying cross-channel weight sharing to discover cross-channel relationships (Parcollet et al., 2018; Parcollet, Morchid, and Linarès, 2019; Gaudet and Maida, 2018, 2021). Also, implementations in hypercomplex space provide more advantages (Arjovsky, Shah, and Bengio, 2016; Danihelka et al., 2016; Hirose and Yoshida, 2012; Nitta, 2002). It has also been shown that the HCNNs could create better output representations (Nitta, 2002; Shahadat and Maida, 2021, 2023).

Recently, HCNNs with various dimensions like 2D HCNNs (Xin, Zhang, and Shao, 2020), 4D HCNNs (quaternion HCNNs) (Gaudet and Maida, 2018; Parcollet et al., 2018; Parcollet, Morchid, and Linarès, 2019), 8D HCNNs (Wu et al., 2020), or HCNNs with arbitrary dimensions (Gaudet and Maida, 2021), have been studied. The reason behind the success of HCNNs is that they capture cross-channel relationships (Parcollet et al., 2018; Parcollet, Morchid, and Linarès, 2019; Gaudet and Maida, 2018, 2021; Shahadat and Maida, 2021). Among them, quaternion networks have a set of algebraic operations and have outperformed other HCNNs. Stacked quaternion convolutional layers produce better representational feature maps and show promising results in vision tasks (Parcollet et al., 2018; Gaudet and Maida, 2018; Shahadat and Maida, 2021). These networks are cost-effective compared to real-valued CNNs and fully connected networks. But still, are very expensive for large inputs like vision tasks.

This work uses a separable hypercomplex network that: 1) handles multidimensional inputs; 2) applies weight sharing across input channels; 3) captures cross-channel correlations; 4) reduces computational costs; and 5) increases validation accuracy performance for image classification datasets. The main idea of this work is to decompose the quaternion convolutional operation into two consecutive separable vectormap convolutional operations, splitting 2D spatial filters into two separable filters with sizes of 3×1 and 1×3. These separable filters are applied to the height and width axis of inputs. It enables the model to reduce costs significantly. Additionally, we apply a quaternion-based stem layer, and parameterized hypercomplex multiplication (PHM) based fully connected layer to get better representation and better generalization performance.

This paper presents the results of extensive experiments that show the effectiveness of separable operations in hypercomplex networks on four image classification datasets. Our contributions are:

- Replacing the spatial 3×3 QCNN in the bottleneck block of quaternion ResNets with two separable VCNNs and showing the improvement of the modified networks.
- Replacing the real-valued CNN with a quaternion-valued

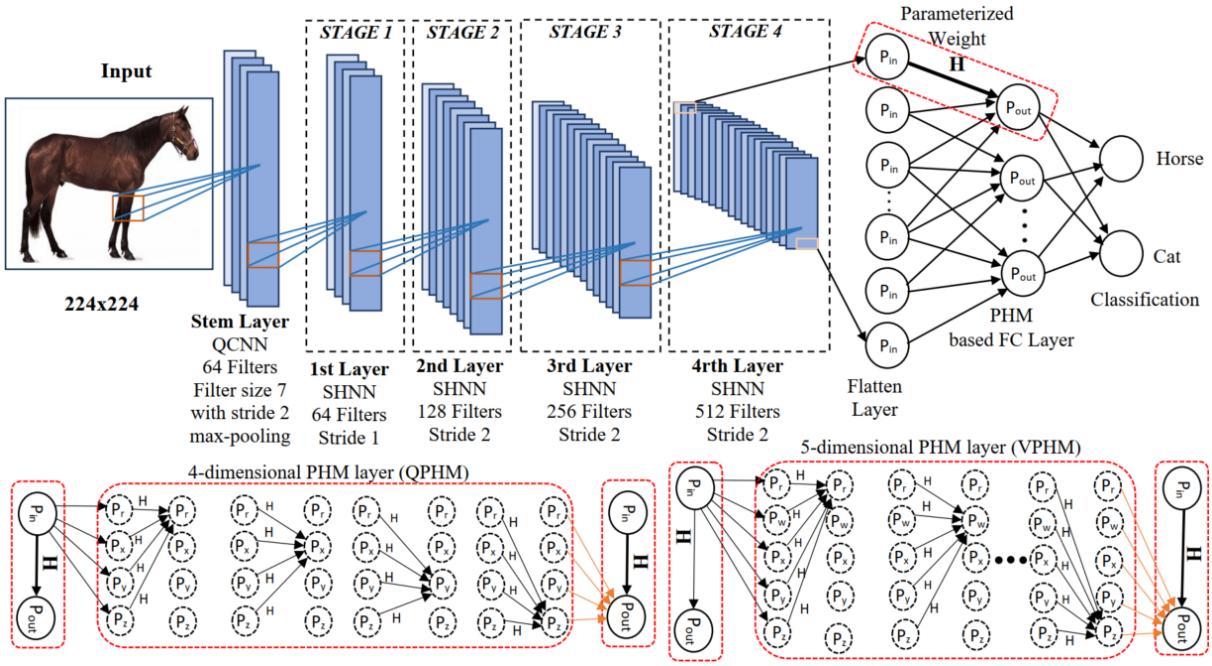

Figure 1: Proposed separable-hypercomplex network with PHM-based fully-connected layer in the backend. "SHNN" stands for separable hypercomplex neural network bottleneck block described in Figure 2. Here, $Q_{in} = Q_r + Q_w + Q_x + Q_y + Q_z$, $H = H_r + H_w + H_x + H_y + H_z$, and $Q_{out} = Q_{ro} + Q_{wo} + Q_{xo} + Q_{yo} + Q_{zo}$ are the input, hypercomplex parameterized weight, and output, respectively. For the calculation of H see the "PHM Layer" section.

CNN in the stem layer (the first layer of the network), resulting in a quaternion-stem model showing performance improvements. Previous quaternion CNNs did not use a quaternion stem.

- Like QPHM (Shahadat and Maida, 2023), applying PHM-based dense layer in the backend of the network.

This proposed separable hypercomplex ResNets outperformed the baseline networks for classification datasets shown in Tables 1 and 2. Our experiments show that the proposed model achieves state-of-the-art results with far fewer trainable parameters, and FLOPS for CIFAR benchmarks in hypercomplex space.

Background and Related Work

Quaternion Convolution

Deep quaternion CNN extends complex CNNs (Trabelsi et al., 2017). This section explains cross-channel weight sharing. (Gaudet and Maida, 2018) and (Parcollet et al., 2018) extended the principles of quaternion convolution operations and weight initialization. A quaternion number system is formed as, $Q = r + ix + jy + kz \; ; \; r, x, y, z \in \mathbb{R}$ where, r, x, y, and z are real values and i, j, and k are imaginary. Quaternion convolution between quaternion filter matrix F and quaternion input vector M, is defined as (Gaudet and

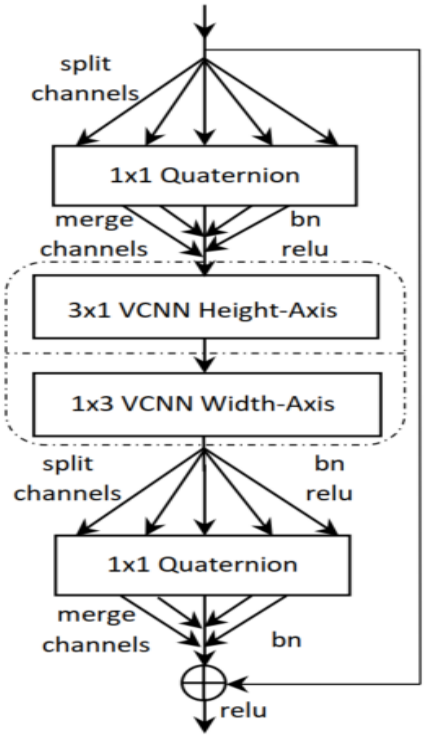

Figure 2: Bottleneck block used in our proposed separable-hypercomplex networks. "bn" and "VCNN" stand for batch normalization and vectormap CNN, respectively.

Maida, 2018):

$$
\begin{aligned}
M \circledast F = (\mathbf{O_r}, \mathbf{O_i}, \mathbf{O_j}, \mathbf{O_k}) = (\\
&\mathbf{M_r} * \mathbf{F_r} - \mathbf{M_i} * \mathbf{F_i} - \mathbf{M_j} * \mathbf{F_j} - \mathbf{M_k} * \mathbf{F_k}, \\
&\mathbf{M_i} * \mathbf{F_r} + \mathbf{M_r} * \mathbf{F_i} + \mathbf{M_j} * \mathbf{F_k} - \mathbf{M_k} * \mathbf{F_j}, \\
&\mathbf{M_j} * \mathbf{F_r} + \mathbf{M_r} * \mathbf{F_j} + \mathbf{M_k} * \mathbf{F_i} - \mathbf{M_i} * \mathbf{F_k}, \\
&\mathbf{M_k} * \mathbf{F_r} + \mathbf{M_r} * \mathbf{F_k} + \mathbf{M_i} * \mathbf{F_j} - \mathbf{M_j} * \mathbf{F_i})
\end{aligned} \tag{1}
$$

where, $\mathbf{M} \circledast \mathbf{F}$, and all others are quaternion numbers. $\mathbf{O_r}$ is the real part, and $\mathbf{O_i}$, $\mathbf{O_j}$, and $\mathbf{O_k}$ are the imaginary parts. Although there are 16 real-valued convolutions in Equation 1, there are only four kernels that are reused. The weight sharing happens this way (Parcollet, Morchid, and Linarès, 2019) and forces the model to learn cross-channel interrelationships. A quaternion layer can accept four or m numbers of input channels, where m is divisible by four. To process m input channels ($m \geq 4$), $m/4$ independent quaternion convolution modules is required. Also, there are $m/4$ weight sets where each module has its own weight set. Cross-channel weight sharing allows discovery of cross-channel input correlations. Our weight initialization was the same as (Gaudet and Maida, 2018).

Vectormap Convolution

We explain 3D generalized hypercomplex CNNs or vectormap CNNs (VCNNs) (Gaudet and Maida, 2021) as VCNNs are used in our proposed models. The VCNN is more flexible as it doesn't require 4D. However, still using cross channel weight sharing this is seen in 3×3 matrix used in Equation 2, only three filters A, B, and C are used. The Vectormap convolution operation is defined as:

$$
\begin{bmatrix} \mathcal{R}(M * F) \\ \mathcal{I}(M * F) \\ \mathcal{J}(M * F) \end{bmatrix} = L \odot \begin{bmatrix} A & B & C \\ C & A & B \\ B & C & A \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{2}
$$

where, A, B, and C are real-valued kernels, x, y, and z being real-valued vectors, and L is a learnable matrix, $L \in \mathbb{R}^{D_3 \times D_3}$; where D_3 stands for 3-dimensional input channels. The initial value of this matrix L is defined as:

$$
L = \begin{bmatrix} 1 & 1 & 1 \\ -1 & 1 & 1 \\ -1 & 1 & 1 \end{bmatrix} \tag{3}
$$

Our weight initialization follows (Gaudet and Maida, 2021).

PHM Layer

Parameterized hypercomplex multiplication is another form of generalized hypercomplex network, explained in (Zhang et al., 2021). As we use this PHM layer only in the fully connected (FC) layer, our explanation is restricted to this PHM-based dense layer. It is defined as, $y = Hx + b$, where $H \in \mathbb{R}^{k \times d}$ represents the PHM layer and it is calculated as, $H = \sum_{i=1}^{n} \mathbf{I}_i \otimes \mathbf{A}_i$, where $\mathbf{I}_i \in \mathbb{R}^{n \times n}$ and $\mathbf{A}_i \in \mathbb{R}^{k/n \times d/n}$ are learnable parameter matrices and $i = 1 \ldots n$ ($n = 4$ or 5). Also, d and k are the input and output dimensions. These matrices can be reused which leads to parameter reduction. Also, the $\otimes$ represents the Kronecker product. The flattened layer which is the output of the CNN network is used as input to the PHM FC layer. These inputs are split as, $Q_{in} = Q_r + Q_w + Q_x + Q_y + Q_z$ and the outputs are merged into Q_{out} as, $Q_{out} = Q_{ro} + Q_{wo} + Q_{xo} + Q_{yo} + Q_{zo}$ for 5D hypercomplex. The 4D hypercomplex parameter matrix is discussed in (Zhang et al., 2021) which expresses the Hamiltonian product, and the 5D hypercomplex parameter matrix of PHM operation is explained in (Shahadat and Maida, 2023). This 5D parameter matrix is used to construct a 5D PHM FC layer which preserves all properties of the PHM layer and hypercomplex networks. This work uses a 5D PHM layer.

Proposed Separable Hypercomplex Networks

Complex CNNs, quaternion CNNs, octonion CNNs, vectormap CNNs, and PHMs are all versions of HCNNs that provide weight sharing across input channels, resulting in the ability to discover cross-channel correlations. These HCNNs perform better, requiring fewer trainable parameters for vision applications. But they are still computationally expensive. For vision tasks, these HCNNs take $\mathcal{O}(N \cdot k^2)$ resources for an image of length N where N is the flattened pixel set. And k is the kernel size. For a 2D image of height h and width w, where $N = hw$, and $h = w$, the computational cost is $\mathcal{O}(h \cdot w \cdot k^2) = \mathcal{O}(h^2 k^2)$.

This section describes our proposed separable hypercomplex model in Figures 1 and 2 to reduce the costs. Separable networks have been used for real-valued networks (Dong et al., 2020; Gholami et al., 2018). Our proposed model follows the assumption that images are approximately square where the pixel count of h and w are the same, and both are much less than the pixel count of hw (Wang et al., 2020). To translate a quaternion convolutional bottleneck block to a separable hypercomplex bottleneck block, we replace the 3×3 spatial quaternion convolutional operation with two separable VCNN layers. These layers are applied sequentially to the height axis (3 channels 3x1 VCNN layer) and width axis (3 channels 1x3 VCNN layer). This decomposition reduces cost from k^2 to $2k$. The two 1×1 quaternion convolutional layers remain unchanged like the original QCNNs (Gaudet and Maida, 2018). The 1×1 QCNNs are used to reduce and then increase the number of channels. This forms our proposed separable hypercomplex bottleneck block seen in Figure 2. This block is stacked multiple times to construct separable hypercomplex ResNets (SHNNs).

Separable hypercomplex models only work on one dimension at a time but the input images are 2-dimensional. For two-dimensional vision tasks, a square 2D input is split into height and width axis. The 3-channel VCNN operation is first applied along the height axis of the input image and then along the width axis of the input image. These two operations finally merge together reducing cost to $\mathcal{O}(h \cdot 2k) = \mathcal{O}(2hk)$ from the HCNN cost of $\mathcal{O}(h^2 k^2)$.

Each quaternion convolution accepts four channels of input and produces four channels of output. Hence, the required number of 1×1 quaternion conv2d modules equals the number of input channels divided by four. The set of output channels of down-sampled 1×1 quaternion is input to the separable VCNN modules, and the output channels of separable VCNN modules are split into groups of

four again for 1×1 the up-sampled quaternion conv2d layer (Gaudet and Maida, 2018; Shahadat and Maida, 2021). One quaternion 2D convolution is applied to each group of four channels and one vectormap 2D convolution is applied to each group of three channels. Like vectormap, each separable vectormap module takes three input channels. Thus, the weight-sharing is compartmentalized into groups of four input channels and then groups of three input channels.

For better representation, a quaternion convolution layer is also used in the stem layer (first layer of the network) as a quaternion-based frontend layer and the fully-connected dense layer as a PHM-based backend layer of deep separable-hypercomplex networks (SHNNs). These have not be used in most previous HCNNs. Figure 1 illustrates our proposed separable-hypercomplex network architecture.

Experiment

We conducted an extensive experiment on four classification datasets to assess the performance of our proposed separable-hypercomplex model. As QCNNs, VCNNs, residual networks (ResNets), QPHM (Shahadat and Maida, 2023), and VPHM (Shahadat and Maida, 2023) all perform 2D spatial convolution operation, we compare our proposed separable hypercomplex networks performance with these baseline models. All models perform Hamiltonian products like our proposed model except for ResNets.

Method

Our experiments used a five-dimensional PHM dense layer in the backend of the network, quaternion network at the beginning of the network, and separable-hypercomplex residual bottleneck block on the CIFAR benchmark datasets (Krizhevsky, Hinton, and others, 2009), the Street View House Numbers (SVHN) (Netzer et al., 2011), and the Tiny ImageNet (Le and Yang, 2015) datasets.

The models we tested to compare with our proposed model, are: the standard deep CNNs (He et al., 2016), the deep QNNs (Gaudet and Maida, 2018), ResNet with QPHM (Shahadat and Maida, 2023), QPHM (Shahadat and Maida, 2023), VPHM (Shahadat and Maida, 2023), and our proposed method. CIFAR-10 and CIFAR-100 datasets consist of $60,000$ color images of size 32×32 pixels. These datasets fall into 10 and 100 distinct classes and are split into a training set with $50,000$ images and a test set with $10,000$ images. We perform standard data augmentation schemes for these datasets like (He et al., 2016; Gaudet and Maida, 2018, 2021; Shahadat and Maida, 2023). Both datasets were normalized using per-channel mean and standard deviation. We perform horizontal flips and take random crops from images padded by 4 pixels on each side to obtain a 40×40 pixel image, then a 32×32 crop is randomly extracted.

SVHN contains about $600,000$ digit images (Netzer et al., 2011). For experiments on SVHN, we don't do any image preprocessing except simple mean/std normalization. We use similar augmentation for the Tiny ImageNet dataset which contains $100,000$ training images of 200 classes (500 per class) downsized to 64×64 color images. The test set has $10,000$ images (Le and Yang, 2015).

All baseline models were trained using the same components as the real-valued networks, the original quaternion network, the original vectormap network, the QPHM, and the VPHM networks using the same datasets. All models in Table 1 were trained using the same hyperparameters and the same number of output channels.

In the stem layer, the 3×3 convolution network is used for deep ResNets (He et al., 2016), 3×3 quaternion network is used for the deep quaternion ResNets (Trabelsi et al., 2017; Gaudet and Maida, 2018), for the QPHM (Shahadat and Maida, 2023), and separable-hypercomplex networks (our proposed method), and 3×3 vectormap network is used for the deep vectormap ResNets (Gaudet and Maida, 2021), and the VPHM (Shahadat and Maida, 2023) networks with stride 1 & 120 output filters. We use parameterized hypercomplex multiplication (PHM) for the dense layer in the backend of deep ResNets, QPHM, VPHM, and our proposed separable-hypercomplex networks. In the bottleneck block, the output channels of bottleneck groups are 120, 240, 480, & 960 for all networks. In this experiment, we analyze 26-layer, 35-layer, and 50-layer architectures with the bottleneck block multipliers "[1, 2, 4, 1]", "[2, 3, 4, 2]", and "[3, 4, 6, 3]".

All of the models were trained using a stochastic gradient descent optimizer. We used linearly warmed-up learning from zero to 0.1 for the first ten epochs and then used a cosine learning rate schedule from 11 to 150. All models were trained using 128 batch sizes.

Results

The results of all models (base models and our proposed networks) appear in Tables 1 and 2. Table 1 shows the results for the CIFAR10, CIFAR100, and SVHN datasets. All datasets were tested using the 26, 35, and 50-layer architectures. The performance measures are the parameter count, FLOPS count (number of multiply-add operations), inference time or latency (time required to process a single image after training), and the percentage accuracy of validation results. We evaluated the original ResNets (He et al., 2016), ResNet with QPHM (Shahadat and Maida, 2023), original quaternion networks (Gaudet and Maida, 2018), original vectormap networks (Gaudet and Maida, 2021), QPHM (Shahadat and Maida, 2023), and VPHM (Shahadat and Maida, 2023) with the same configuration like our proposed separable-hypercomplex networks. Our proposed separable-hypercomplex networks perform almost 3% better in validation accuracy with fewer parameters and FLOPS for CIFAR-10, CIFAR-100, and SVHN datasets than the baseline networks. More precisely, our proposed method takes almost six times, three times, two times, three times, and two times fewer parameters than the ResNets, quaternion networks, vectormap networks, QPHM, and VPHM respectively. Moreover, separable-hypercomplex networks achieved state-of-the-art results for these CIFAR benchmarks in hypercomplex space.

The performances for the Tiny ImageNet datasets are shown in Table 2 for all architectures. The separable-hypercomplex network's validation accuracies outperform the other base networks with fewer trainable parameters and

Model Name	Layers	Params	FLOPS	Latency	Validation Accuracy		
					CIFAR10	CIFAR100	SVHN
ResNet	26	41.2M	2.56G	0.89ms	94.68	78.21	96.04
RQPHM		40.9M	2.56G	0.64ms	95.32	79.14	96.64
Quaternion		10.6M	1.15G	0.64ms	94.89	77.65	95.88
Vectormap		13.6M	1.15G	0.64ms	94.76	77.65	95.93
QPHM		10.3M	1.11G	0.65ms	95.26	78.15	95.97
VPHM		13.4M	1.09G	0.66ms	95.15	78.14	96.24
SHNN		**6.2M**	**1.06G**	**0.69ms**	**95.91**	**79.42**	**97.21**
ResNet	35	58.1M	3.31G	1.07ms	94.95	78.72	95.74
RQPHM		57.8M	3.31G	0.81ms	95.80	79.65	96.22
Quaternion		14.5M	1.51G	0.81ms	95.33	78.96	95.95
Vectormap		19.3M	1.48G	0.84ms	95.06	79.52	95.97
QPHM		14.5M	1.47G	0.82ms	95.55	78.46	95.99
VPHM		19.6M	1.45G	0.82ms	95.60	79.86	96.34
SHNN		**9.2M**	**1.36G**	**0.85ms**	**96.49**	**79.93**	**97.25**
ResNet	50	82.9M	4.57G	1.36ms	94.08	78.95	95.76
RQPHM		82.6M	4.57G	1.09ms	95.86	79.89	96.78
Quaternion		21.09M	1.96G	1.06ms	95.42	79.17	96.24
Vectormap		27.6M	1.93G	1.13ms	95.37	79.39	96.39
QPHM		20.7M	1.93G	1.05ms	95.75	78.22	96.46
VPHM		27.5M	1.92G	1.08ms	95.76	79.49	96.49
SHNN		**13.6M**	**1.75G**	**1.09ms**	**96.79**	**80.81**	**97.47**

Table 1: Image classification performance on the CIFAR benchmarks and SVHN for 26, 35, and 50-layer architectures. QPHM, VPHM, and RQPHM define the quaternion networks with the PHM FC layer, vectormap networks with the PHM FC layer, and ResNets with the PHM FC layer, respectively.

Model Name	Layers	Parameters	FLOPS	Latency	Accuracy
ResNet	26	41.6M	10.2G	3.06	57.21
RQPHM		41M	2.56G	2.31	57.84
Quat.		11.0M	4.54G	2.48	53.84
Vect.		14.4M	4.56G	2.88	56.15
QPHM		10.4M	1.11G	2.31	54.02
VPHM		13.8M	4.44G	3.27	53.11
SHNN		**6.3M**	**1.06G**	**2.49**	**58.56**
ResNet	35	58.5M	13.2G	3.21	57.80
RQPHM		57.9M	3.31G	2.85	59
Quat.		15.2M	5.98G	3.52	54.53
Vect.		20.0M	5.98G	3.76	55.99
QPHM		14.6M	1.47G	2.88	56.42
VPHM		19.4M	5.88G	4.08	56.10
SHNN		**9.3M**	**1.36G**	**2.97**	**60.06**
ResNet	50	83.2M	18.2G	3.77	59.06
RQPHM		82.6M	4.57G	3.66	60.30
Quat.		21.4M	7.87G	4.14	56.63
Vect.		28.3M	7.87G	4.34	57.52
QPHM		20.8M	1.93G	3.88	59.42
VPHM		27.7M	7.75G	4.51	58.96
SHNN		**13.7M**	**1.75G**	**3.93**	**62.73**

Table 2: Image classification performance on the Tiny ImageNet benchmarks for 26, 35, and 50-layer architectures. Here, "Quat.", "Vect.", "RQPHM", "QPHM", and "VPHM" stand for quaternion networks, vectormap networks, ResNets with QPHM, the quaternion networks with the PHM FC layer, and vectormap networks with the PHM FC layer, respectively.

FLOPS like CIFAR benchmarks and SVHN datasets. The result Tables 1, and 2 show our proposed model performance ranges of three runs. However, the latency of separable-hypercomplex networks is a little bit higher in some cases than the quaternion-based networks. This may be due to the use of vectormap networks along with quaternion networks

as the latency for vectormap networks is higher.

Discussion and Conclusions

This paper proposes separable-*hypercomplex* convolutions to reduce the cost of 2D convolutional operations and shows their effectiveness of image classification tasks. We also applied 4D PHM in the network's backend. On CIFAR benchmarks, our proposed separable-hypercomplex network was formed by stacking separable vectormap convolutions (three-dimensional) in the quaternion bottleneck blocks, achieved state-of-the-art results among hypercomplex networks.

Our main conclusion is that using quaternion convolutions as the frontend stem layer, four/five-dimensional PHM-based densely connected backend layer, and separable-hypercomplex bottleneck block improves classification performance on the CIFAR benchmarks, SVHN, and Tiny ImageNet datasets in comparison to the other models we tested. Our proposed method factorizes a channel-wise 2D convolution (hypercomplex convolution which works along the channels) to a column convolution and a row convolution. Extensive experiments show that this leads to systematic improvement with far fewer trainable parameters on image classification. This proposed method can save almost 36% and 51% trainable parameters compared to original quaternion and vectormap networks and QPHM and VPHM networks, respectively.

Although our proposed separable-hypercomplex design reduced parameter counts and FLOPS, it exhibited higher latency than real-valued and hypercomplex-valued CNNs. This is because the model performs convolution twice (height-axis and width-axis) and takes transition time from quaternion to two separable VCNNs. As we replaced spatial quaternion (four-dimensional hypercomplex network) 2D convolution using two separable vectormap (three-dimensional hypercomplex network) convolutions, the number of output channels is restricted to 120 or a multiple of 120 which are divisible by three and four. Our investigation concludes that the performance comparison between the hypercomplex networks and our proposed separable-hypercomplex networks shows that the separable-hypercomplex convolution provides better validation performance with fewer trainable parameters and FLOPS for image classification tasks.

Further work may be directed toward the architecture of the separable quaternion network and separable vectormap network. Moreover, other datasets will be tested to check if these proposed architectures can perform similarly or not. Finally, separable-quaternion and separable vectormap convolutional methods will help to remove the number-of-channels constraint.

References

Arjovsky, M.; Shah, A.; and Bengio, Y. 2016. Unitary evolution recurrent neural networks. In *International Conference on Machine Learning*, 1120–1128. PMLR.

Buyssens, P.; Elmoataz, A.; and Lézoray, O. 2012. Multiscale convolutional neural networks for vision–based classification of cells. In *Asian Conference on Computer Vision*, 342–352. Springer.

Danihelka, I.; Wayne, G.; Uria, B.; Kalchbrenner, N.; and Graves, A. 2016. Associative long short-term memory. In *International Conference on Machine Learning*, 1986–1994. PMLR.

Dong, N.; Zhao, L.; Wu, C.-H.; and Chang, J.-F. 2020. Inception v3 based cervical cell classification combined with artificially extracted features. *Applied Soft Computing* 93:106311.

Gaudet, C. J., and Maida, A. S. 2018. Deep quaternion networks. In *2018 International Joint Conference on Neural Networks (IJCNN)*, 1–8. IEEE.

Gaudet, C. J., and Maida, A. S. 2021. Removing dimensional restrictions on complex/hyper-complex neural networks. In *2021 IEEE International Conference on Image Processing (ICIP)*, 319–323. IEEE.

Gholami, A.; Kwon, K.; Wu, B.; Tai, Z.; Yue, X.; Jin, P.; Zhao, S.; and Keutzer, K. 2018. Squeezenext: Hardware-aware neural network design. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1638–1647.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hirose, A., and Yoshida, S. 2012. Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence. *IEEE Transactions on Neural Networks and learning systems* 23(4):541–551.

Javanmardi, S.; Ashtiani, S.-H. M.; Verbeek, F. J.; and Martynenko, A. 2021. Computer-vision classification of corn seed varieties using deep convolutional neural network. *Journal of Stored Products Research* 92:101800.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Le, Y., and Yang, X. S. 2015. Tiny imagenet visual recognition challenge.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Nitta, T. 2002. On the critical points of the complex-valued neural network. In *Proceedings of the 9th International Conference on Neural Information Processing, 2002. ICONIP'02.*, volume 3, 1099–1103. IEEE.

Parcollet, T.; Ravanelli, M.; Morchid, M.; Linarès, G.; Trabelsi, C.; De Mori, R.; and Bengio, Y. 2018. Quaternion recurrent neural networks. *arXiv preprint arXiv:1806.04418*.

Parcollet, T.; Morchid, M.; and Linarès, G. 2019. Quaternion convolutional neural networks for heterogeneous image processing. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8514–8518. IEEE.

Shahadat, N., and Maida, A. S. 2021. Adding quaternion representations to attention networks for classification. *arXiv preprint arXiv:2110.01185*.

Shahadat, N., and Maida, A. S. 2023. Enhancing resnet image classification performance by using parameterized hypercomplex multiplication. *arXiv preprint arXiv:2301.04623*.

Trabelsi, C.; Bilaniuk, O.; Zhang, Y.; Serdyuk, D.; Subramanian, S.; Santos, J. F.; Mehri, S.; Rostamzadeh, N.; Bengio, Y.; and Pal, C. J. 2017. Deep complex networks. *arXiv preprint arXiv:1705.09792*.

Wang, H.; Zhu, Y.; Green, B.; Adam, H.; Yuille, A.; and Chen, L.-C. 2020. Axial-deeplab: Stand-alone axial-attention for panoptic segmentation. In *European Conference on Computer Vision*, 108–126. Springer.

Wu, J.; Xu, L.; Wu, F.; Kong, Y.; Senhadji, L.; and Shu, H. 2020. Deep octonion networks. *Neurocomputing* 397:179–191.

Xin, R.; Zhang, J.; and Shao, Y. 2020. Complex network classification with convolutional neural network. *Tsinghua Science and technology* 25(4):447–457.

Zhang, A.; Tay, Y.; Zhang, S.; Chan, A.; Luu, A. T.; Hui, S. C.; and Fu, J. 2021. Beyond fully-connected layers with quaternions: Parameterization of hypercomplex multiplications with $1/n$ parameters. *arXiv preprint arXiv:2102.08597*.