Road Crack Detection Using Quaternion Neural Networks

Aggelos Katsaliros[*], Iason-Ioannis Panagos[*], Giorgos Sfikas[*†‡], Christophoros Nikou[*]

[*]Department of Computer Science and Engineering, University of Ioannina, Ioannina, Greece
[†]Department of Surveying and Geoinformatics Engineering, University of West Attica, Athens, Greece
[‡]Computational Intelligence Laboratory, Institute of Informatics and Telecommunications,
National Center for Scientific Research "Demokritos", Greece
aggeloskatsaliros@outlook.com, i.panagos@uoi.gr, gsfikas@uniwa.gr, cnikou@uoi.gr

***Abstract*—Automatic road crack detection can play an significant role in improving driving safety. However, a number of factors make this task challenging. Background complexity, as well as the fact that cracks are visually inhomogenous to one another also contribute to this difficulty. Furthermore, cracks in road surfaces can be easily confused with foreign objects, shadows and background textures leading to detection ambiguities. Timely detection of cracks is important for both drivers and road maintenance crews. Quaternion neural networks (QNNs) are a relatively new class of neural networks that employ quaternion-valued activations and parameters. They benefit from reduced costs in terms of hardware as they require fewer parameters. In this work, we explore the usability of QNNs for automatic road crack detection. To this end, we propose quaternionic versions of deep networks and evaluate their performance in datasets with images of road cracks. We show that the proposed models are light-weight in terms of parameter requirements while they are on par in terms of performance with real-valued networks for crack detection, with potential applications in resource constrained scenarios or in cases where few training data is available.**

***Keywords*—Quaternions, Road Crack Detection, Hypercomplex Numbers**

I. Introduction

Road crack detection is a complex task which involves detecting irregularities and damage in road surfaces such as concrete, cement, pavement or asphalt. It is a multi-faceted task which also requires classification and segmentation in order to produce accurate results. Due to their nature, cracks in surfaces involve many low-level features and visual patterns and are thus hard to detect and classify. When designing accurate and robust models for road crack detection, one needs to take into account additional factors that make this process challenging. Such factors include sub-par image quality which often appears in the form of variances in illumination, low resolution, or blurring, as well as visual noise, for example from stains, shadows, reflections, or litter on the road surface. Applications for road crack detection can be found in real life problems, such as automatic car driving safety, automated road inspection and also in ensuring the quality of transport services.

Detecting road cracks (in a non-automatic way) is a time-consuming and potentially dangerous job that requires a significant amount of time, many hours of manpower, specialized equipment as well as trained road crews and engineers. Progress in computer vision and machine learning has enabled researchers to develop automatic crack detection methods that produce good results. Nowadays, most methods take advantage of the effectiveness of representative features provided by deep convolutional neural networks (CNNs) for road crack detection in a supervised classification manner [1], [2]. Road crack detection is a time-critical task, as the time between detection and repair can be long, resulting in accidents that could otherwise be prevented. In addition, prolonged exposure to the elements could further deteriorate damage to the road surface, requiring longer time to repair as well as increased costs. A potential drawback of deep convolutional methods is their running speed and cost of resources, preventing them from being utilized in real-time scenarios or in portable devices and equipment available to road maintenance personnel.

Quaternions are hypercomplex numbers that are comprised of 4 components: one real and three imaginary. Their three imaginary components make them ideal for image processing since each one of these components can embed a color part of an RGB image. Lately, research interest in quaternions has increased with many proposed models outperforming their real-valued equivalents in various tasks such as image processing [3], [4] and speech recognition [5]. Additionally, due to the interactions of the Hamilton product, quaternion-valued networks benefit from parameter sharing, which results in lightweight models that require fewer parameters and less storage space. Quaternionic layers can be used in place of traditional (real-valued) ones to provide these advantages to deep architectures, reducing their size without significant loss of performance.

In this work, we compare the results of image classification and segmentation CNN architectures with their respective quaternionic counterparts in the task of road crack detection. Specifically, we replace standard layers with quaternion-valued versions and compare the accuracy, precision and number of parameters required by the new models. A reduced-data training regime is devised, in which new datasets are produced by sampling from a large source dataset of road crack images to assess network performance in conditions where training data is not readily available.

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call "RESEARCH - CREATE - INNOVATE", project *Culdile* (code T1EΔK-03785).

The remainder of this paper is structured as follows: In Section II we present related work in the areas of road crack detection with deep convolutional features, as well as recent research on quaternions. Elements and properties of quaternions are overviewed in Section III. We provide an analysis of our proposed method in Section IV. Experimental setups, datasets, evaluation metrics, and results are presented in Section V. Finally, conclusions are drawn in Section VI.

II. Related Work

A. Road Crack Detection with CNNs

The success of convolutional neural networks (CNNs) in many computer vision tasks has led to researchers developing various architectures for road crack detection. For instance, [6] developed a CNN architecture, which classifies image patches that contain cracks. The central pixels of a patch determine whether it is regarded as positive (contains a crack) or negative. More recently, the authors of [1] combined a deep CNN for crack detection with thresholding for crack extraction from images that are classified as containing cracks. Their method consists of two steps: first, a CNN is used to determine the presence of cracks in an image, and then, for the positive samples, adaptive thresholding takes place to segment and extract the cracks. In [2] an end-to-end deep hierarchical CNN that predicts pixel-wise crack segmentation called *DeepCrack* is proposed. In contrast to the typical approaches to feature extraction which occurs at the last convolutional layer of the network, *DeepCrack* aggregates features from multiple scales and levels while a deeply-supervised net is used for feature supervision which is applied at each convolutional stage. The outputs are then fused, and the final predictions are obtained by guided filtering.

B. Quaternion Neural Networks

Quaternion neural networks (QNNs) utilize quaternion-valued layers and inputs instead of real-valued ones. Even though research in quaternion representations on deep learning is relatively new, a few works have been proposed that explore their applications. More specifically, [7] proposed deep quaternion networks for classification as well as segmentation. Their results show that quaternions achieve competitive performance for these tasks and they require fewer parameters. [4] developed Quaternion Convolutional Neural Networks (QCNNs) to better represent colored images in the quaternion domain. They test their QCNN models for color image classification and denoising and compare them to traditional CNNs achieving better results. In [8], the authors focus on the effect of the Hamilton product on color image reconstruction with gray-scale only training. To that end, they develop a quaternion convolutional encoder-decoder architecture for reconstruction of a unique gray-scale image. Compared to a traditional convolutional encoder-decoder network, their model demonstrates the ability to successfully learn to reconstruct the image colors while trained only on the gray-scale version of an image. They conclude that quaternion-valued architectures are not hampered by internal and global dependencies and are suitable for applications in image recognition. The same authors propose Quaternion Recurrent Neural Networks (QRNNs) for sequential tasks such as speech recognition [3]. They show that their quaternion-based recurrent architectures outperform non-quaternion versions while requiring 2 to 3 times fewer parameters. Sfikas et al. [9] extended generative adversarial networks (GANs) with quaternions for text detection in the wild. The proposed network performs on par with a real-valued GAN, while being much less expensive in terms of model size which has potential applications in resource-constrained scenarios.

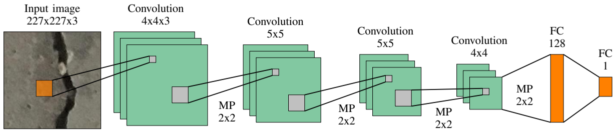


Fig. 1: Architecture of the CNN used as baseline. FC stands for Fully-Connected, and MP for Max-Pooling.

III. Quaternions

Quaternions are hypercomplex numbers, introduced by Hamilton in 1843 as extensions of complex numbers to four dimensions. Each quaternion $q \in \mathbb{H}$ consists of 1 real and 3 imaginary parts and can be represented as:

$$q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}, \tag{1}$$

where $a, b, c, d \in \mathbb{R}$ and $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are independent imaginary units that obey the following multiplication rules:

$$\begin{gathered} \mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1 \\ \text{and} \\ \mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \end{gathered} \tag{2}$$

From these relations one can observe that multiplication is non-commutative for quaternions. Other fundamental relations on $\mathbb{H}$ are defined as follows, for two quaternions p, q where $p = a_p + b_p\mathbf{i} + c_p\mathbf{j} + d_p\mathbf{k}$ and $q = a_q + b_q\mathbf{i} + c_q\mathbf{j} + d_q\mathbf{k}$:

Addition:

$$p + q = (a_p + a_q) + (b_p + b_q)\mathbf{i} + (c_p + c_q)\mathbf{j} + (d_p + d_q)\mathbf{k}. \tag{3}$$

Scalar Multiplication:

$$\lambda q = \lambda a + \lambda b_i + \lambda c_j + \lambda d_k. \tag{4}$$

Conjugation:

$$\bar{q} = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k}. \tag{5}$$

Quaternion magnitude:

$$|q| = \sqrt{q\bar{q}} = \sqrt{\bar{q}q} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{6}$$

In addition, the real part of a quaternion can be expressed as sum of a scalar $S(q)$ and an imaginary part as a three dimensional vector $V(q)$. Thus, the quaternion can be represented as:

$$q = S(q) + V(q) = (a + 0i + 0j + 0k) + (0 + bi + cj + dk). \tag{7}$$

For two quaternions p, q using the above expression, the *Hamilton product* ($\otimes$) can be written as:

$$\begin{aligned} p \otimes q = {} & S(p)S(q) - V(p) \cdot V(q) \\ & + S(p)V(q) + S(q)V(p) + V(p) \times V(q) \end{aligned} \tag{8}$$

TABLE I: Accuracy and F-score comparison for standard and quaternion-valued (Q) models, after training on the various dataset sizes, evaluated on the concrete images dataset [10]. “RS“ stands for Reduced Size (see Section IV, §3).

Method	Metric	Training dataset size								Parameters
		28K	21K	14K	7K	3.5K	1.75K	0.7K	0.35K	
AlexNet [11]	Accuracy	0.989	0.982	0.942	0.935	0.931	0.741	<0.5	<0.5	58.285.441
	F-Score	0.985	0.979	0.937	0.932	0.928	0.634	<0.5	<0.5	
AlexNet (Q)	Accuracy	0.997	0.989	0.991	0.987	0.981	0.971	0.840	<0.5	14.584.513
	F-Score	0.996	0.987	0.990	0.985	0.978	0.968	0.803	<0.5	
VGG-16 [12]	Accuracy	0.997	0.996	0.996	0.992	0.984	0.984	0.966	0.959	33.609.793
	F-Score	0.994	0.994	0.993	0.990	0.982	0.977	0.956	0.959	
VGG-16 (Q)	Accuracy	0.994	0.994	0.994	0.989	0.983	0.971	0.968	0.943	8.421.313
	F-Score	0.991	0.993	0.993	0.987	0.977	0.968	0.962	0.932	
Proposed	Accuracy	0.997	0.993	0.991	0.989	0.978	0.968	0.908	0.51	4.747.489
	F-Score	0.996	0.992	0.990	0.988	0.976	0.963	0.890	0.640	
Proposed (Q)	Accuracy	0.996	0.992	0.995	0.991	0.985	0.971	0.932	<0.5	1.187.553
	F-Score	0.995	0.992	0.993	0.988	0.982	0.966	0.921	<0.5	
Proposed (RS)	Accuracy	0.994	0.990	0.992	0.990	0.976	0.978	<0.5	<0.5	1.188.037
	F-Score	0.992	0.990	0.992	0.987	0.973	0.977	<0.5	<0.5	

where $\cdot$ is the dot product and $\times$ is the cross product. An equivalent formulation of the Hamilton product is the matrix form:

$$\begin{aligned} p \otimes q = &(a_p a_q - b_p b_q - c_p c_q - d_p d_q) \\ &+ (a_p b_q + b_p a_q + c_p d_q - d_p c_q)\mathbf{i} \\ &+ (a_p c_q - b_p d_q + c_p a_q + d_p b_q)\mathbf{j} \\ &+ (a_p d_q + b_p c_q - c_p b_q + d_p a_q)\mathbf{k}. \end{aligned} \tag{9}$$

The Hamilton product replaces the standard real valued dot product; in case the real parts of p and q are zero, it boils down to simply a cross product.

Quaternion convolutional neural networks [4], [8] are composed of quaternionic parameters, inputs, activation functions and outputs. An RGB image can represented in the quaternion domain as:

$$q(p_{x,y}) = 0 + R(p_{x,y})\mathbf{i} + G(p_{x,y})\mathbf{j} + B(p_{x,y})\mathbf{k}. \tag{10}$$

If o^l_{ab} and S^l_{ab} are the quaternion output and pre-activation quaternion output respectively at layer l and (a, b) are the indexes of the new feature map, for a $K \times K$-sized quaternion-valued weight filter map w, the quaternion convolution can be described as:

$$o^l_{ab} = s(S^l_{ab}) \tag{11}$$

with

$$S^l_{ab} = \sum_{c=0}^{K-1} \sum_{d=0}^{K-1} w^l \otimes o^{l-1}_{(a+c)(b+d)}, \tag{12}$$

where s is the activation split function. The split activation function is defined as:

$$s(q) = f(a) + f(b)\mathbf{i} + f(c)\mathbf{j} + f(d)\mathbf{k}, \tag{13}$$

where f can be any standard (real-valued) activation function.

Quaternion-valued networks share weights through the Hamilton product, which allows them to learn the internal relations of the features of a quaternion, effectively encoding the component information. Thus, models using quaternion-valued layers can benefit from parameter savings since a quaternion weight that links two quaternion units has 4 degrees of freedom, compared to a standard neural net that has 4×4 and requires more parameters as a result [3].

IV. Deep Networks For Crack Detection

Baseline model We propose a real-valued CNN architecture (shown in Figure 1) based on [6] to be used as a baseline for our experiments. It consists of 4 convolutional and max pooling layers, followed by 2 linear layers and a sigmoid activation function. The total number of parameters of this model are 4.747.489.

Proposed quaternion-valued variants We replace the convolution and linear layers with their quaternion-valued equivalents in our proposed model as well as in popular CNN architectures from the literature. In the quaternion models we convert the RGB image into a quaternion matrix in which every element is a quaternion that contains the RGB values of the pixel in its imaginary part. As a result, when convolution between quaternion matrix and kernel is performed, the Hamilton product will produce an output which is a quaternion with RGB values in the imaginary part [4]. At the output of the last quaternionic linear layer the 4 channels are summed.

As discussed briefly in Section III and in [3], using quaternion-valued layers leads to a parameter reduction to about $1/4$ of the original amount. Due to the amount of training data with respect to the number of parameters of the real-valued architectures, overfitting is possible, especially in cases of limited available data. Therefore, to make the performance comparison more accurate, we modify the number of channels in our proposed model to obtain a real-valued reduced size version with a comparable number of parameters to its quaternion equivalent.

V. Experiments and Analysis

A. Datasets

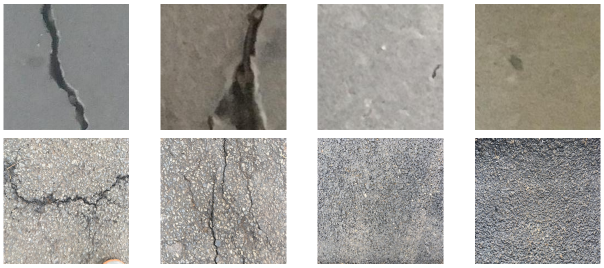
Training: For a more robust evaluation of the proposed method, different dataset sizes were used for training. We use [10], which contains images of concrete of a resolution of $227 \times 227 \times 3$ (height, width and channels), as a source to generate new datasets of reduced size. The original dataset is divided in two sets, as negative and positive crack images, each containing 20.000 samples. Figure 2 (top row) depicts some examples from both classes. By randomly reducing the number of samples, while keeping the number of crack and non-crack images equal we generate 8 new reduced-size datasets, named after the total number of training samples, to be used for

TABLE II: Performance comparison of standard and quaternionic (Q) models on the asphalt dataset [13] under different training sizes. Accuracy and F-scores are shown.

Method	Metric	Training dataset size								Parameters
		28K	21K	14K	7K	3.5K	1.75K	0.7K	0.35K	
AlexNet [11]	Accuracy	0.795	0.762	0.747	0.820	0.835	0.782	<0.5	<0.5	58.285.441
	F-Score	0.737	0.783	0.728	0.645	0.801	0.716	<0.5	<0.5	
AlexNet (Q)	Accuracy	0.647	0.575	0.675	0.755	0.615	0.675	0.720	<0.5	14.584.513
	F-Score	<0.5	<0.5	<0.5	0.652	<0.5	<0.5	0.585	<0.5	
VGG-16 [12]	Accuracy	0.770	0.722	0.787	0.712	0.830	0.835	0.802	0.772	33.609.793
	F-Score	0.661	0.520	0.717	0.560	0.754	0.806	0.801	0.765	
VGG-16 (Q)	Accuracy	0.682	0.745	0.703	0.817	0.772	0.812	0.767	0.850	8.421.313
	F-Score	0.513	0.607	0.530	0.786	0.682	0.746	0.765	0.824	
Proposed	Accuracy	0.802	0.837	0.851	0.795	0.797	0.832	0.760	<0.5	4.747.489
	F-Score	0.727	0.746	0.803	0.715	0.736	0.784	0.652	0.659	
Proposed (Q)	Accuracy	0.655	0.660	0.740	0.810	0.857	0.790	0.830	0.770	1.187.553
	F-Score	0.436	0.463	0.593	0.743	0.843	0.719	0.788	0.687	
Proposed (RS)	Accuracy	0.807	0.817	0.857	0.742	0.712	0.637	0.595	0.545	1.188.037
	F-Score	0.738	0.758	0.797	0.629	0.657	<0.5	<0.5	<0.5	

training our models. A sample split of 70%-15%-15% for training, validation and testing respectively is maintained for all new datasets. The produced datasets can be seen on Table III. This training regime allows us to assess the performance of our models in conditions where training data is limited.

Fig. 2: Positive and negative samples from [10] (top row) and [13] (bottom row).



Testing: We test our models in the concrete [10] and asphalt [13] road crack datasets. [13] contains 400 images taken from asphalt, 200 for each class (containing cracks for positive, and without a crack for negative). The initial size of $448 \times 448 \times 3$ is resized to $227 \times 227 \times 3$.

TABLE III: Reduced-size datasets produced from [10] for assessing network performance under different numbers of available training samples. K indicates 1.000 and positive means that the image contains a crack.

Samples	28K	21K	14K	7K	3.5K	1.75K	0.7K	0.35K
Train Positive	14.000	10.500	7.000	3.500	1.750	875	350	175
Val Positive	3.000	2.250	1.500	750	375	190	75	40
Train Negative	14.000	10.500	7.000	3.500	1.750	875	350	175
Val Negative	3.000	2.250	1.500	750	375	190	75	40

B. Evaluation Metrics

Let n_{ij} be the number of pixels of the class i predicted to be the class j, where n_{cls} is the number of different classes and $t_i = \sum_j n_{ij}$ is the total number of pixels of the class i (both true and false positives). The following metrics are used for evaluating the networks:
Global Accuracy (G), which measures the percentage of pixels that have been predicted correctly: $\frac{\sum_i n_{ii}}{\sum_i t_i}$, Precision (P): $\frac{TP}{TP+FP}$, Recall (R): $\frac{TP}{TP+FN}$ and F-Score (F): $\frac{2PR}{R+R}$, where TP, FP and FN are the true positives, false positives and false negatives, respectively.

C. Training Setup

We evaluate the performance of various standard and quaternionic models after training for 10 epochs on the aforementioned reduced size datasets. All models are trained from scratch with SGD using a learning rate of 0.01. The batch size is set to 10. The experiments are performed on a machine with a Nvidia Titan Xp GPU.

D. Analysis of results

We report maximum global accuracy and F-score results on the concrete [10] and asphalt [13] test sets on Tables I and II respectively. We note that using quaternionic layers leads to a parameter reduction of about 75%. A general trend observed is that reducing the dataset size lowers the accuracy and F-scores for all networks, especially when tested on the asphalt dataset. However, in both datasets, the quaternion-valued versions retain acceptable performance even under a very limited amount of training data, and in some cases even perform better, while requiring fewer parameters. This indicates that quaternionic networks can function better with smaller datasets, avoiding overfitting, and thus can be viable alternative to standard ones when there is not enough data.

VI. Conclusion

Timely detection and repair of road damage is essential for the safety of drivers and passengers. In this work, we explored the applicability of quaternion-valued layers in deep CNNs for the task of road crack detection. We proposed quaternion models that are able to accurately detect cracks in images and perform on par with real-valued ones while requiring significantly fewer parameters. In fact, utilizing all quaternionic layers leads to a parameter reduction of 75%, compared to using standard layers. We conclude that quaternion-valued networks are a promising alternative to real-valued ones, since they are able to effectively reduce model size without lowering performance, even when trained on a very low amount of samples. Consequently, quaternionic models are ideal for deployment in scenarios where hardware requirements are low or the available data are scarce. As future work, we aim to further research recent developments in hypercomplex models [14] and their application to crack detection.

References

[1] R. Fan, M. J. Bocus, Y. Zhu, J. Jiao, L. Wang, F. Ma, S. Cheng, and M. Liu, "Road crack detection using deep convolutional neural network and adaptive thresholding," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 474–479.

[2] Y. Liu, J. Yao, X. Lu, R. Xie, and L. Li, "Deepcrack: A deep hierarchical feature learning architecture for crack segmentation," *Neurocomputing*, vol. 338, pp. 139–153, 2019.

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

[4] X. Zhu, Y. Xu, H. Xu, and C. Chen, "Quaternion convolutional neural networks," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 631–647.

[5] X. Qiu, T. Parcollet, M. Ravanelli, N. Lane, and M. Morchid, "Quaternion neural networks for multi-channel distant speech recognition," *arXiv preprint arXiv:2005.08566*, 2020.

[6] L. Zhang, F. Yang, Y. D. Zhang, and Y. J. Zhu, "Road crack detection using deep convolutional neural network," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3708–3712.

[7] C. J. Gaudet and A. S. Maida, "Deep quaternion networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

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

[9] G. Sfikas, A. Giotis, G. Retsinas, and C. Nikou, "Quaternion Generative Adversarial networks for inscription detection in byzantine monuments," in *2nd International Workshop on Pattern Recognition for Cultural Heritage (PatReCH)*, 2021.

[10] C. Ozgenel, "Concrete crack images for classification, mendeley data, v2, doi: 10.17632/5y9wdsg2zt.2," 2018. [Online]. Available: https://data.mendeley.com/datasets/5y9wdsg2zt/2

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[13] A. Jayanth Balaji, G. Thiru Balaji, M. S. Dinesh, N. Binoy, and D. S. Harish Ram, "Asphalt crack dataset, mendeley data, v2, doi: 10.17632/xnzhj3x8v4.2," 2019. [Online]. Available: https://data.mendeley.com/datasets/xnzhj3x8v4/2

[14] I.-I. Panagos, G. Sfikas, and C. Nikou, "Compressing audio visual speech recognition models with parameterized hypercomplex layers," in *Hellenic Conference on Artificial Intelligence (SETN)*, 2022.