A Quaternion Deterministic Monogenic CNN Layer for Contrast Invariance

Eduardo Ulises Moya-Sánchez, Sebastià Xambó-Descamps, Sebastián Salazar Colores, Abraham Sánchez Pérez, and Ulises Cortés

Abstract Deep learning (DL) is attracting considerable interest as it currently achieves remarkable performance in many branches of science and technology. However, current DL cannot guarantee capabilities of the mammalian visual systems such as lighting changes. This paper proposes a deterministic entry layer capable of classifying images even with low-contrast conditions. We achieve this through an improved version of the quaternion monogenic wavelets. We have simulated the atmospheric degradation of the CIFAR-10 and the Dogs and Cats datasets to generate realistic contrast degradations of the images. The most important result is that the accuracy gained by using our layer is substantially more robust to illumination changes than nets without such a layer.

E. U. Moya-Sánchez (✉) · A. Sánchez Pérez
Gobierno de Jalisco, Guadalajara, Mexico
e-mail: eduardo.moya@jalisco.gob.mx

A. Sánchez Pérez
e-mail: abraham.sanchez@jalisco.gob.mx

E. U. Moya-Sánchez
Universidad Autónoma de Guadalajara, Zapopan, Mexico

S. Xambó-Descamps · U. Cortés
UPC-BarcelonaTech, Barcelona, Spain
e-mail: sebastia.xambo@upc.edu

U. Cortés
e-mail: ia@lsi.upc.edu

Barcelona Supercomputing Center, Barcelona, Spain

S. Salazar Colores
Centro de investigación en Óptica, León, Mexico
e-mail: sebastiansalazar@cio.mx

S. Xambó-Descamps (ed.), *Systems, Patterns and Data Engineering with Geometric Calculi*, SEMA SIMAI Springer Series 13, https://doi.org/10.1007/978-3-030-74486-1_7

512467_1_En_7_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:12/5/2021 Pages: xxx Layout: T1-Standard

Acronyms

CIFAR	Canadian Institute for Advanced Research
CNN	Convolutional Neural Network
DL	Deep Learning
FC	Fully Connected Layer
FL	Flatten Layer
FME	Facultat de Matemàtiques i Estadística
GA	Geometric Algebra
GC	Geometric Calculus
HSV	Hue-Saturation-Value (color standard)
MCNN	Monogenic CNN
ML	Machine Learning
MP	Max pooling layer
NN	Neural Network
RESNET	Residucal Network
RGB	Read-Green-Blue (color standard)
UPC	Universitat Politìcnica de Catalunya

1 Introduction

Many authors argue that currently, no suitable theory of CNNs design is available [1, 2]. Although some evidence supports that the depth (number of layers) [3], and the data augmentation during the training process [4], can occasionally provide invariance or equivariance relative to some class of transformations, the reasons for that behaviour do not seem to be well understood. Some investigations indicate that the learning of an invariance response may fail even with very deep CNNs or by large data augmentations in the training [1].

To overcome these shortcomings one idea is to embrace suitable geometric methods, as in [5], where the main techniques are real algebraic varieties and methods of computer algebra, and in [1, 2, 6–10], in which methods of differential geometry are used. In this regard, another strategy to progress in "Geometric Deep Learning" is to use Geometric Calculus (GC) in the sense of [11–14]. The main strong points for this advance are the long history of achievements in a great variety of fields (see [12], §6.4, and the references therein); that it includes the complex numbers and the quaternions as exceptional cases; and the fact that there is a well-developed theory of GC wavelets with the potential to be applied to DL much as scalar wavelets are used in current DL techniques [15–17]. An additional bonus of GC is that the representation of the signals occurs in a higher-dimensional space, and hence they provide a more robust discrimination capacity naturally.

In this work, as the first step in this general strategy, we work with Hamilton's quaternions **H**, which is the most straightforward geometric calculus beyond the

complex numbers **C** (see Appendix A). The main results are the design and implementation of a CNN layer (M6) based on the *monogenic signal* proposed by Feslberg et al. [18]. These layers substantially enhance the invariance response to illumination-contrast.

Up till now, quaternions have been used with fully connected NNs [19–22], and, more recently, with CNNs [23–26]. In this context, the method proposed in this paper is the first, to the best of our knowledge, that combines a CNN with local phase computations using quaternions.

On the experimental side, to evaluate the predictive performance of M6, we have simulated illumination-contrast changes using the atmospheric degradation model (see Appendix B) over two image-datasets, the CIFAR-10 [27] and the Dogs and Cats [28].

The rest of the paper is organised into four additional sections and two appendices. The core of the paper consists of Sects. 2 and 3, which describe the new monogenic layer, M6, and the experimental setup, respectively. The experiments, results and analyses are presented in Sect. 4. Our conclusions are drawn in Sect. 5. Finally, in appendixes A and B, we summarise what we need about the quaternion field **H** and about the atmospheric scattering model of light, respectively.

2 Monogenic Convolution Neural Network Layer

What we call Monogenic Convolutional Neural Network (MCNN) is the coupling of a deterministic layer based on the monogenic signal, which we call *monogenic layer* (M6), with a conventional convolutional neural network (CNN). What we accomplish in this way, as shown by the experiments in subsequent sections, is a system for classifying images that not only outperforms the usual CNNs in speed but which is also resilient in front of severe changes in contrast.

2.1 Monogenic Signal

The terminology we are going to use is as follows (cf. Felsberg et al. [18]). We define 1D (resp. 2D) *multivectorial signals* as C^1 maps $U \to \mathcal{G}$ from an interval $U \subset \mathbf{R}$ (a region $U \subset \mathbf{R}^2$) into a *geometric algebra* $\mathcal{G}$ (see [12]). For $\mathcal{G} = \mathbf{R}$ ($\mathcal{G} = \mathbf{C}$, $\mathcal{G} = \mathbf{H}$) we say that the signal is *scalar* (*complex, quaternionic*). For technical reasons we also assume that signals are in L^2 (that is, the modulus is square-integrable).

The *Riesz-Felsberg transform* (RF) maps 2D scalar signals to 2D quaternionic signals. Among the signals obtained in this way, our interest lies in the (quaternionic) *monogenic signals* (see [18] for details). Some applications of the monogenic signal are: visual perception measurements [29, 30], local feature detection such as lines (even-signal) and edges (odd-signal) [14, 31], estimation of the disparity of stereo

images and blending of images [32], or computation of fast phase-based video magnification [33].

The monogenic signal $I_M = I_M(x, y) \in \mathbf{H}$ associated to an image[1] $I = I(x, y) \in \mathbf{R}$ (where $x, y \in U$, U a region of $\mathbf{R}^2$). The definition of I_M is as follows (cf. [18]):

$$I_M = I + I_{M'}, \quad I_{M'} = \boldsymbol{i}\, I_1 + \boldsymbol{j}\, I_2, \tag{1}$$

where, denoting by $*$ the convolutional product,

$$\begin{aligned} I_1 &= I * h_1, \quad I_2 = I * h_2, \\ h_1(x, y) &= -\tfrac{x}{2\pi(x^2+y^2)^{3/2}}, \\ h_2(x, y) &= -\tfrac{y}{2\pi(x^2+y^2)^{3/2}}. \end{aligned} \tag{2}$$

The signals I_1 and I_2 are the *Riesz transforms* (quadrature filters) of I in the x and y directions [18]. Note that $I_M \in \langle 1, \boldsymbol{i}, \boldsymbol{j} \rangle \subset \mathbf{H}$.

The *local amplitude signal* $|I_M|$ is defined by $|I_M|(x, y) = |I_M(x, y)|$, where the last expression is the modulus of the quaternion $I_M(x, y)$ [18]. Notice that we have

$$|I_M|^2 = |I|^2 + |I_{M'}|^2 = |I|^2 + |I_1|^2 + |I_2|^2, \tag{3}$$

where $|I|(x, y) = |I(x, y)|$ and similarly with $|I_1|$ and $|I_2|$.

2.2 Monogenic Filter Bank

In practice the monogenic signal needs a bandpass filtering, in order to define the local region of the signal [14, 34]. For all the computations of the filtered version of the image I we have used a radial (isotropic) bandpass *Log-Gabor* function in the frequency domain $G(u_1, u_2)$) defined as follows:

$$G(u_1, u_2) = \exp\left(-\frac{\log\left(\frac{\sqrt{u_1^2+u_2^2}}{\omega_0} \right)^2}{2\log(\sigma)^2} \right), \tag{4}$$

where u_1, u_2 are frequency components, ω_0 is the central frequency of the filter, σ is a bandwidth parameter (see [35] for more details). The filtering process is described in the following steps:

[1] Here it is to be noted that I is not the source image we are interested in, but a filtered version of it in the sense explained in Sect. 3.

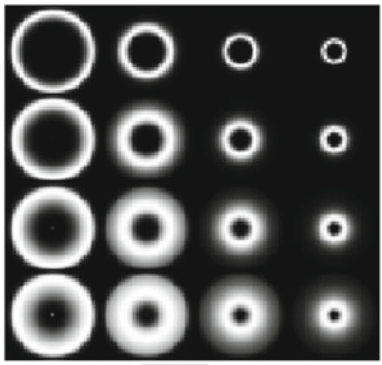


Fig. 1 **a** Log-Gabor filters at different scales and bandwidths in the frequency domain

1. Compute the 2D Fourier transform $\mathcal{F}(\bar{J})$ of the mean value image $\bar{J}$ as in [36], namely

$$\mathcal{F}(\bar{J})[u_1, u_2] = \sum_{m_1} \sum_{m_2} \bar{J}[m_1, m_2] e^{-i2\pi(u_1 m_1 + u_2 m_2)} \tag{5}$$

2. We have modified the P. Kovesi Python implementation [37], in order to compute a monogenic filter bank based on $G(u_1, u_2)$ with different scales. A filter bank can be computed with the following parameters $\omega_0^s = \frac{1}{\min_{wl} * s_f^{s-1}}$, where $\min_{wl}$ is the minimum wavelength, s_f is a scale factor, $s = 1, 2, \ldots, n_s$ is the current scale.

$$\bar{J}^s(u_1, u_2) = \bar{J}(u_1, u_2) \exp\left(- \frac{\log\left(\frac{\sqrt{u_1^2 + u_2^2}}{\omega_0^s} \right)^2}{2 \log(\sigma)^2} \right) \tag{6}$$

Figure 1 presents various views of the Log-Gabor function $G(u_1, u_2)$.

The *local phase* I_ϕ and the *local orientation* I_θ associated to I are defined, following [18], by the relations

$$I_\phi = \operatorname{atan2}\left(\frac{I}{|I_R|} \right), \tag{7}$$

$$I_\theta = \operatorname{atan}\left(\frac{-I_2}{I_1} \right), \tag{8}$$

where the quotients of signals are taken point-wise. For the geometric interpretation of these signals see Fig. 2.

The local phase can distinguish lines and edges [14, 20, 30, 31, 38], whereas the local orientation appears as a pinwheel picture resembling the behaviour of V1 simple cells and orientation columns.

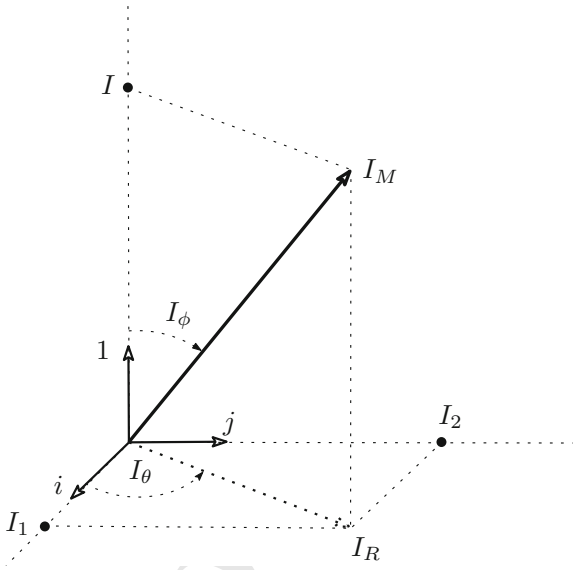


Fig. 2 Geometry of the monogenic signal

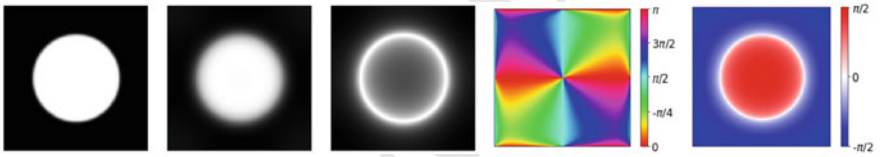


Fig. 3 Original image, filtered image, local energy I_M, local phase I_ϕ, and local orientation I_θ

Figure 3 illustrates the monogenic transform of the image of a white circle. From left to right, we display the original image, the filtered image (in the sense of Sect. 3), and the corresponding local magnitude, phase and orientation signals. The highest local energy values take place at the circle boundary, whereas the dominant values of the local orientation are $-\pi/2$ (blue), 0 (white), $\pi/2$ (red).

2.3 *Description of the Monogenic Layer*

We report on work about one monogenic layer, which we call M6. As we will see, both share promising features with the V1 layer in which they are inspired.

The purpose of M6 is to perform the following set of operations on the input image J. If J is formed by different channels indexed by $c = 0, 1, \ldots, N-1$, we denote by J_c the image corresponding to channel c. For instance, in an RGB (HSV) image, we would have the images I_R, I_G, and I_B (I_H, I_S, and I_V).

1. Get the average $\bar{J}$ of J over the channels forming J, that is, with the previous notations,

$$\bar{J} = \frac{1}{N}\sum_{c=0}^{N-1} J_c \tag{9}$$

2. Get the image I obtained by filtering $\bar{J}$ in the sense explained in Sect. 3.
3. Calculate the monogenic components $|I_M|$, I_ϕ, I_θ of I as defined by the Eqs. (3), (7), and (8), respectively.
4. Construct two HSV images as follows:

$$HSV_\phi = (H, S, V) = (I_\phi, |I_R|, 1), \tag{10}$$

$$HSV_\theta = (H, S, V) = (I_\theta, |I_R|, 1). \tag{11}$$

5. Transform the HSV images into RGB images RGB_ϕ and RGB_θ according to the standard conventions (see page 304 of [39])

The use of the colour space HSV has been handy for the encoding in different colour hues the values of the monogenic angular components. Moreover, the further transformation to the RGB colour space enhances the visibility of the regions in the original image in which the local amplitude is significant, which translates into a sensitivity to sharp edges.

The six components of $M6$, namely RGB_θ plus RGB_ψ, together with the 3 RGB components of the input signal J, are illustrated in Fig. 4.

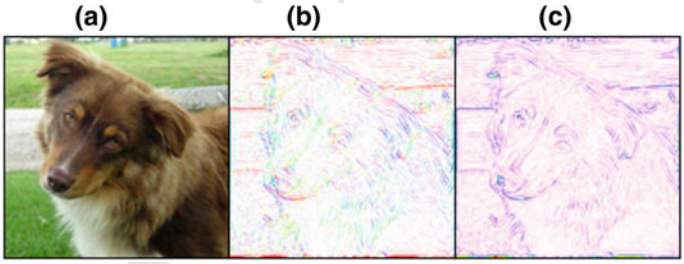


Fig. 4 **a** RGB input image. **b** RGB_θ. **c** RGB_ϕ. $M6$ is defined by **b** and **c**

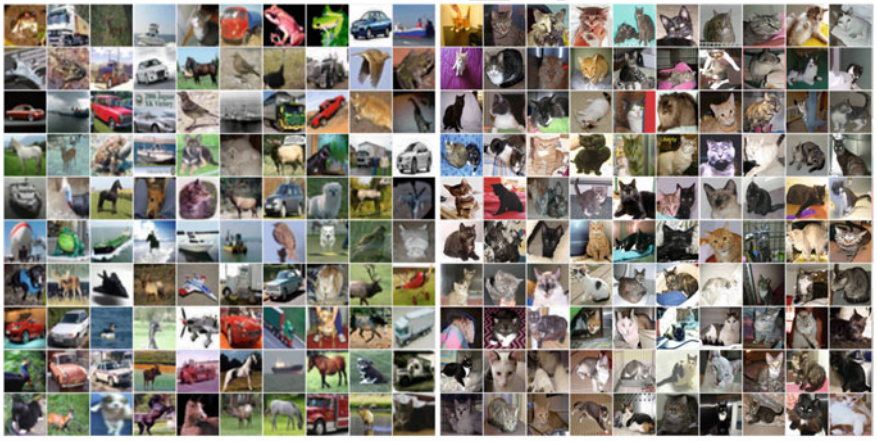
3 Experimental Setup

3.1 Datasets

We have used two datasets, CIFAR-10 [27] (Fig. 5(a)) and Dogs and Cats (Fig. 5(b)). Each data set was split into three sets: training set, validation set and test set. Table 1 shows their main characteristics.

Table 1 Characteristics of the CIFAR-10 and Dogs and Cats datasets

	CIFAR-10	Cats vs Dogs
Training set	36,000	2,400
Validation set	12,000	800
Test set	12,000	800
Total	60,000	4,000
Input shape	[32 × 32 × 3]	[150 × 150 × 3]



(a) (b)

Fig. 5 Examples of: **a** 100 RGB images from the CIFAR-10 dataset; **b** 100 images from the Dogs and Cats dataset

3.2 Degrading Procedures

In our experiments, any of the original images is degraded by the addition of fog according to the McCartney atmospheric scattering model [40] (summarized in Appendix B) and further modified by the addition of independent random changes in the illumination colour (in the range 0.8–1.0 for each colour component) of atmospheric light $A(r, g, b)$. In addition, the transmission map $t(x, y)$ was computed with the random parameters summarized in the Table 2.

An account of the details is deferred to Appendix B, while Fig. 6 provides an illustration of a degradation run of the images in Fig. 5. For concreteness, we consider three degraded levels (d_1, d_2, d_3) for each dataset; for an illustration, see Fig. 7

Table 2 Degradation parameters of colour of atmospheric light $A(r, g, b)$ and the transmission map $t(x, y)$

Degradation levels	
$A([0.8, 1], [0.8, 1], [0.8, 1])$	
Level	Parameters
d_0	Zero degradation
d_1	$t(x, y) = [0.5, 0.8]$
d_2	$t(x, y) = [0.3, 0.5]$
d_3	$t(x, y) = [0.0, 0.15]$

(a) (b)

Fig. 6 Degraded versions (d_1) of the images in Fig. 5: **a** From CIFAR-10; **b** From Dogs and Cats

Fig. 7 **a** Original image, level d_0; **b** contrast level d_1; **c** contrast level d_2; **d** contrast level d_3. For the meaning of these labels see Subsect. 3.5

3.3 CNN and MCNN Architectures

Functionally, an NN layer takes an input $\mathbf{x}$ and produces an output $\mathbf{x}'$. The map $f : \mathbf{x} \mapsto \mathbf{x}'$ depends on parameters associated with the layer and whose nature depends on the kind of layer. In general, $\mathbf{x}$, $\mathbf{x}'$, and the layer parameters are multidimensional arrays whose nature is chosen according to the processing that has to be achieved.

Write $[n_1, n_2, \ldots, n_d]$ to denote the type of a d-dimensional (real) array with axis dimensions $n_1, \ldots, n_d$. Thus $[n]$ is the type of n-dimensional vectors and $[n_1, n_2]$, the type of matrices with n_1 rows and n_2 columns. Matrices are useful to represent monochrome images, but for RGB images we need arrays of type $[n_1, n_2, 3]$, or $[n_1, n_2, n_3]$ if it is required that the image be represented by n_3 *channels*, as for example $n_3 = 6$ for a pair of color stereoscopic images.

The parameters associated to convolutional and fully connected layers are represented by a *filter* array of weights, W,[2] and a *bias* array, $\mathbf{b}$. In these cases, the expression of f has the form

$$f^{\pi}(\mathbf{x}) = g(\mathbf{x} \star_{\pi} W + \mathbf{b}), \tag{12}$$

where $\star_{\pi}$ is a *pairing* specific of the layer and g is an activation function, ReLU in this paper, that is applied component-wise to arrays. For convolutional layers, $\star_{\pi} = \star$ is *array cross-correlation*, to be described below, while for fully connected layers, $\star_{\pi}$ is matrix product, which is denoted by juxtoposition of its factors, $\mathbf{x}W$. For a maximum pooling layer, the parameters are represented by a triple of positive integers $(w_1, w_2, s = 1)$, where (w_1, w_2) is the shape of the pooling window and s is the stride (1 by default). In this case $\star_{\pi} = \star_{\mathrm{mp}}$ is given by the rule

$$(\mathbf{x} \star_{\mathrm{mp}} W)[i, j, k] = \max(\mathbf{x}[is : is + w_1 - 1, js : js + w_2 - 1, k]), \tag{13}$$

where we use the standard slicing conventions for arrays. The shape of the array $\mathbf{x} \star_{\mathrm{mp}} W$ is $[n'_1, n'_2, n_3]$, where n'_1 and n'_2 are the greatest integers such that $n'_1 \le (n_1 - w_1)/s$ and $n'_2 \le (n_2 - w_2)/s$.

[2] Filters are also called *kernels*.

In the cross-correlation product $\mathbf{y} = \mathbf{x} \star W$, $\mathbf{x}$ is an array of type $[n_1, n_2, n_3]$ and W (the filter) is an array of type $[w_1, w_2, n_3, m_3]$. The pair (n_1, n_2) is the shape of the space dimensions of $\mathbf{x}$ and n_3 the number of channels. The pair (w_1, w_2) denotes the window dimensions of the filter and m_3 the number of channels of the array $\mathbf{y}$. The definition is given by the following formula:

$$\mathbf{y}[i, j, k] = \sum_{m=0}^{w_1-1} \sum_{n=0}^{w_2-1} \sum_{r=0}^{n_3-1} \mathbf{x}[i+m, j+n, r] W[m, n, r, k], \tag{14}$$

which can be expressed more compactly as

$$\mathbf{y}[i, j, k] = \sum_{r=0}^{n_3-1} \mathbf{x}[i : i + w_1 - 1, j : j + w_2 - 1, r] * W[:, :, r, k], \tag{15}$$

where $*$ denotes the ordinary scalar product of matrices. Notice that the shape of $\mathbf{y}$ is $[n_1 - w_1 + 1, n_2 - w_2 + 1, m_3]$.

There is also a *downsampled cross-correlation* $\mathbf{y} = \mathbf{x} \star_s W$ by a stride s:

$$\begin{aligned} \mathbf{y}[i, j, l] &= \sum_{k,m,n} \mathbf{x}[is + m, js + n, k] W[m, n, k, l] \\ &= \sum_{k} \mathbf{x}[is : is + w_1 - 1, js : js + w_2 - 1, k] \\ &\qquad * W[:, :, k, l]. \end{aligned} \tag{16}$$

Table 3 Flow of the monogenic CNN and of CNN-1 for the CIFAR-10 dataset. CNN-1 has 1,250,858 trainable parameters

CIFAR-10		
I	$[32, 32, n_3 = 3]$	
$M6$	$[32, 32, n_3 = 6]$	
$\mathbf{x} \to \mathbf{x}'$	W	$\mathbf{x}'$
C*-1	$[3, 3, n_3, 32]$	[32, 32, 32]
C-2	[3, 3, 32, 32]	[30, 30, 32]
MP-1	$[2, 2, s = 2]$	[15, 15, 32]
C*-3	[3, 3, 32, 64]	[15, 15, 64], Dropout (0.25)
C-4	[3, 3, 32, 64]	[13, 13, 64]
MP-2	$[2, 2, s = 2]$	[6, 6, 64] Dropout (0.25)
FL		[2304]
FC-1	[2304, 512]	[512]
FC-2	[512, 10]	[10] Dropout (0.5)
SMAX		[10]

Table 4 Flow of the monogenic CNN, and of CNN-2, for Cats and Dogs dataset. CNN-2 with 2,797,730 trainable parameters

Cats and Dogs		
I	$[224, 224, n_3 = 3]$	
$M6$	$[224, 224, n_3 = 6]$	
$\mathbf{x} \to \mathbf{x}'$	W	$\mathbf{x}'$
C*-1	$[3, 3, n_3, 32]$	[224, 224, 32]
MP-1	$[2, 2, s = 2]$	[112, 112, 32]
C-2	[3, 3, 32, 32]	[110, 110, 32]
MP-2	$[2, 2, s = 2]$	[55, 55, 32]
C-2	[3, 3, 32, 64]	[53, 53, 64]
MP-2	$[2, 2, s = 2]$	[26, 26, 64]
FL		[43264]
FC-1	[438264, 64]	[64] Dropout (0.5)
SMAX		[2]

The shape of the array $\mathbf{x} \star_s W$ is $[n'_1, n'_2, n_3]$, where n'_1 and n'_2 are the greatest integers such that $n'_1 \leq (n_1 - w_1)/s$ and $n'_2 \leq (n_2 - w_2)/s$.

In this work, we have used two architectures: CNN-1 (for CIFAR-10) and CNN-2 (for Dogs and Cats). See Fig. 8 for a schematic representation of M6CNN-1, which consists of CNN-1 with the M6 layer (M6CNN-2 is defined similarly). The computation flux of these networks is summarized in Table 3 for CIFAR-10 and in Table 4 for Cats and Dogs. In these two tables, the input I is processed by M6, and the resulting output is processed by a sequence of convolutional (C),[3] max-pooling (MP), flatten (FL), fully connected (FC), and softmax (SMAX) layers. If the monogenic step is omitted, the flow agrees with fairly standard CNNs (here called CNN-1 and CNN-2; see below for further details). The value n_3 is equal to 3 for I and 6 for M6, respectively. The W column specifies the filter of the current step. It is to be understood that the action of the layers C and FC is completed with a ReLU activation function.

Initially, we tested our monogenic layer on top of two CNNs architectures, CNN-1 and CNN-2, both with nine hidden layers. These testings aimed to carry out a relatively fast search of adequate hyper-parameter values that guarantee a classification accuracy close to a baseline mark. Additionally, we have tested our layer on top of a well-known RESNET-20 v2 architecture (with 571,034 trainable parameters) [41], with 18 hidden layers, to ascertain that it also produced gains similar to those observed with the simpler architectures.

[3] C* is a convolution with zero padding.

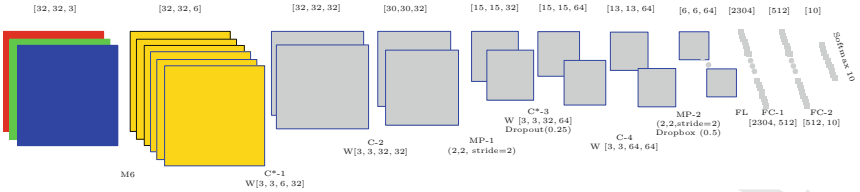


Fig. 8 CNN-1 architecture with 1,250,858 trainable parameters including the monogenic layer, four convolutional layers, 2 dropout, two max polling layers, three fully connected and one softmax function

Table 5 Experimental arrangements for each CNN

CNN	
Trained	Tested
d_0	d_0, d_1, d_2, d_3
d_1	d_0, d_1, d_2, d_3
d_2	d_0, d_1, d_2, d_3
d_3	d_0, d_1, d_2, d_3

3.4 Monogenic Hyper Parameters

To find the best parameters of the Monogenic layer, we evaluated the validation accuracy of CNN-1 on the CIFAR-10 dataset up to 100 epochs, with scales $s = [3, 4, 5]$, minimum wavelength $\min_w = [3, 4, 5]$ (i.e. smallest scale filter), scaling factors $sf = [1.1, 1.2...2.1]$, and standard deviations $\sigma = [0.3, 0.4, 0.48, 0.6]$. Finally we tried the learning rate values $lr = [0.0001, 0.001, 0.005]$. Altogether this amounts to 1584 combinations. The outcome was that the best parameters are $lr = 0.0001$, $s = 1$, $\min_w = 3$, $\sigma = 0.25$, $sf = 1.1$, for a maximum of test accuracy and minimum processing time.

3.5 Experiments

We trained and tested six nets: CNN-1, M6CNN-1, CNN-2, M6CNN-2, RESNET-20 v2, M6-RESNET-20 v2 [41], following the scheme summarized in Table 5, where d_i ($i = 0, 1, 2, 3$) means a degradation degree (see Fig. 7 for an intuitive view of the significance of these values). The computational codes are available at https://github.com/asp1420/monogenic-cnn-illumination-contrast.

In order to test each of all trained models about their generalization capacity, they were run not only on the original test set but also on the three modified versions of it consisting in adding the same three levels of haze.

All the experiments were carried out for 100 epochs, and learning rate of $lr = 0.0001$, no data augmentation, on the CTE-power 9 cluster of the Barcelona Supercomputing Center with one Tesla V-100.

4 Results and Analysis

A synopsis of experimental results for CNN-1 and CNN-2 is reported in Fig. 9 for CIFAR-10 a Fig. 10 for Cats and Dogs. Similarly, Fig. 11 summarizes the findings for RESNET-20 in the case of CIFAR-10.

The general idea of the experimental design has been to train the nets under four different degradation levels. These trained systems are represented on the horizontal axes and labelled by the degradation labels d_j ($j = 0, 1, 2, 3$). Each d_j is run on a set of images not seen before and also presented in four degradation levels d_k ($k = 0, 1, 2, 3$). The observed accuracies are represented by coloured circles in the case of the basic net and by coloured squares in the case of the corresponding monogenic-enhanced net. Thus each of the three graphics quotes 32 accuracies, 16 for the basic net and 16 for the monogenic net.

The main finding is the resilience of each of the monogenic systems d_j with respect to any of the degradations d_k, for the squares above d_j are clustered around 0.70 accuracy for all d_k. This contrasts with the wide dispersion of the circles above d_j, with a (to be expected) maximum when $k = j$ and substantially lower values for $k \neq j$. To note, however, that the basic nets d_j perform slightly better just for the degradation d_j, as shown by the top position of several of the corresponding circles (in Fig. 9, for instance, blue circle for $d_0 - d_0$, green for $d_1 - d_1$, and magenta for $d_2 - d_2$).

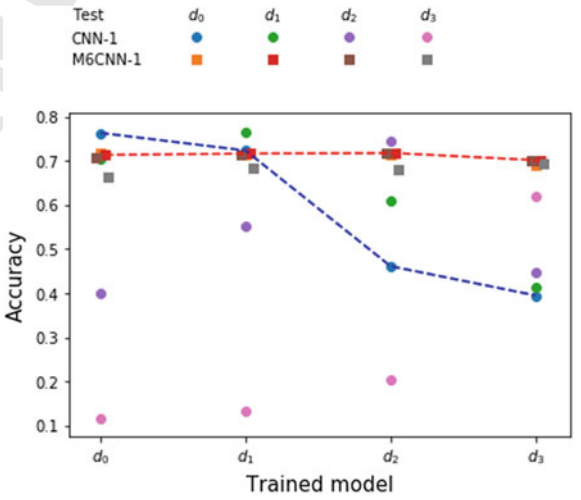


Fig. 9 CIFAR-10 test data with different degradation using CNN1 models. See text for details

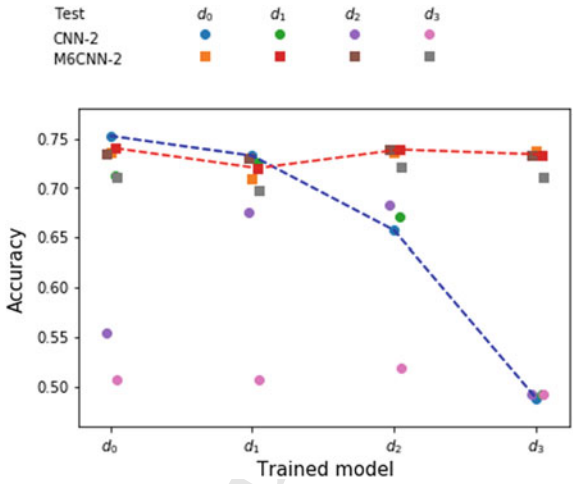


Fig. 10 Dogs and Cats test data with different degradation models using CNN-2. See text for details

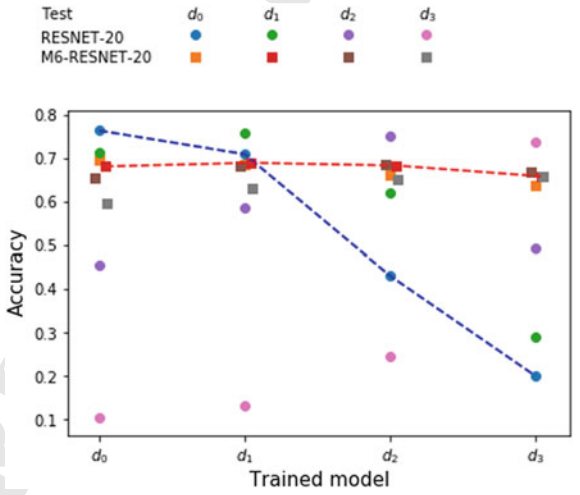


Fig. 11 CIFAR-10 test data with different degradation models using RESNET-20. See text for details

Our experiments results are consistent with the theoretical invariance of the phase-based feature detection to brightness and contrast transformations reported at [38, 42]. The M6 activation maps presented at Fig. 12 strengthened our confidence in that the M6 is invariant to these transforms, due to the activation maps is visually the same even with different levels of degradation.

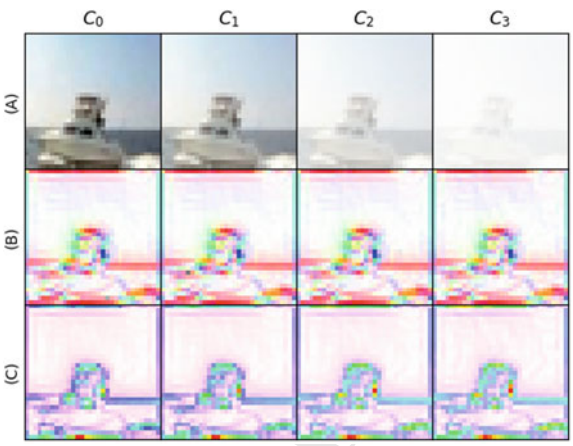


Fig. 12 Activation maps of M6. Row **A** present a one CIFAR-10 image with different degradation levels. Row **B** presents the activation map (local orientation) with different image degradation. Row **C** shows the local phase activation map with different input images

5 Conclusions

The context of this paper is the idea that geometric calculus has the potential to articulate novel and promising researches in deep learning.

In the explorations reported in this paper, we have used the quaternion calculus, which is the most straightforward geometric calculus beyond the complex calculus. More specifically, we have designed a front layer for CNNs that processes a monogenic signal by extracting phase and orientation signals and assembling them in an HSV space.

The experimental results with two different datasets and three CNNs confirm that the accuracy gained by using our layer has a substantially more robust performance when faced with severe illumination changes than the same nets without such a front layer.

We plan to continue the trail walked in this research by developing a front layer for CNNS that is resilient when faced with other transformations of the images, like rotations or even small deformations.

Acknowledgments The authors would like to thank to CONACYT and Barcelona supercomputing Center. Sebastián Salazar-Colores (CVU 477758) would like to thank CONACYT (Consejo Nacional de Ciencia y Tecnología) for the financial support of his PhD studies under Scholarship 285651. Ulises Moya and Ulises Cortés are member of the Sistema Nacional de Investigadores CONACyT.

Appendices

A. Quaternion Algebra

The quaternion algebra $\mathbf{H}$ is a four dimensional real vector space with basis $1, \boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$,

$$\mathbf{H} = \mathbf{R}1 \oplus \mathbf{R}\boldsymbol{i} \oplus \mathbf{R}\boldsymbol{j} \oplus \mathbf{R}\boldsymbol{k} \tag{17}$$

endowed with the bilinear product (multiplication) defined by Hamilton's relations, namely

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{i}\,\boldsymbol{j}\boldsymbol{k} = -1. \tag{18}$$

As it is easily seen, these relations imply that

$$\boldsymbol{i}\,\boldsymbol{j} = -\boldsymbol{j}\boldsymbol{i} = \boldsymbol{k}, \quad \boldsymbol{j}\boldsymbol{k} = -\boldsymbol{k}\boldsymbol{j} = \boldsymbol{i}, \quad \boldsymbol{k}\boldsymbol{i} = -\boldsymbol{i}\boldsymbol{k} = \boldsymbol{j}. \tag{19}$$

The elements of $\mathbf{H}$ are named *quaternions*, and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$, *quaternionic units*. By definition, a quaternion q can be written in a unique way in the form

$$q = a + b\boldsymbol{i} + c\,\boldsymbol{j} + d\boldsymbol{k}, \quad a, b, c, d \in \mathbf{R}. \tag{20}$$

Its *conjugate*, $\bar{q}$, is defined as

$$\bar{q} = a - (b\boldsymbol{i} + c\,\boldsymbol{j} + d\boldsymbol{k}). \tag{21}$$

Note that $(q + \bar{q})/2 = a$, which is called the *real part* or *scalar part* of q, and $(q - \bar{q})/2 = q - a = b\boldsymbol{i} + c\,\boldsymbol{j} + d\boldsymbol{k}$, the *vector part* of q.

Since the conjugates of $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are $-\boldsymbol{i}, -\boldsymbol{j}, -\boldsymbol{k}$, the relations (18) and (19) imply that the conjugation is an *antiautomorphism* of $\mathbf{H}$, which means that it is a linear automorphism such that $\overline{qq'} = \bar{q}'\bar{q}$.

Using Hamilton's relations again, we easily conclude that

$$q\bar{q} = a^2 + b^2 + c^2 + d^2. \tag{22}$$

This allows to define the *modulus* of q, $|q|$, as the unique non-negative real number such that

$$|q|^2 = q\bar{q}. \tag{23}$$

Observe that $|qq'| = |q||q'|$. Indeed, $|qq'|^2 = qq'\overline{qq'} = qq'\bar{q}'\bar{q} = q|q'|^2\bar{q} = |q|^2|q'|^2$.

Finally, for $q \neq 0$, $|q| > 0$ and $q(\bar{q}/|q|^2) = 1$, which shows that any non-zero quaternion has an inverse and therefore that $\mathbf{H}$ is a (skew) field.

B. The Atmospheric Scattering Model

We have used the atmospheric scattering model in order to model the illumination and contrast degradation of the images. The formation of a degraded image is modeling using the atmospheric scattering model proposed by McCartney et al. [40], defined as follows:

$$I(x, y) = J(x, y)t(x, y) + A(1 - t(x, y)), \tag{24}$$

where $I(x, y)$ is the measured image, $J(x, y)$ is the original scene without affectations, $A(r, g, b)$ is the illumination colour of atmospheric light, and $t(x, y)$ is named as transmission map, which can be defined in a homogeneous atmosphere as:

$$t(x, y) = e^{-\beta d(x,y)}, \tag{25}$$

where β is the scattering coefficient of the atmosphere and $d(x, y)$ is the scene depth. An example of the estimated transmission map is presented in Fig. 13.

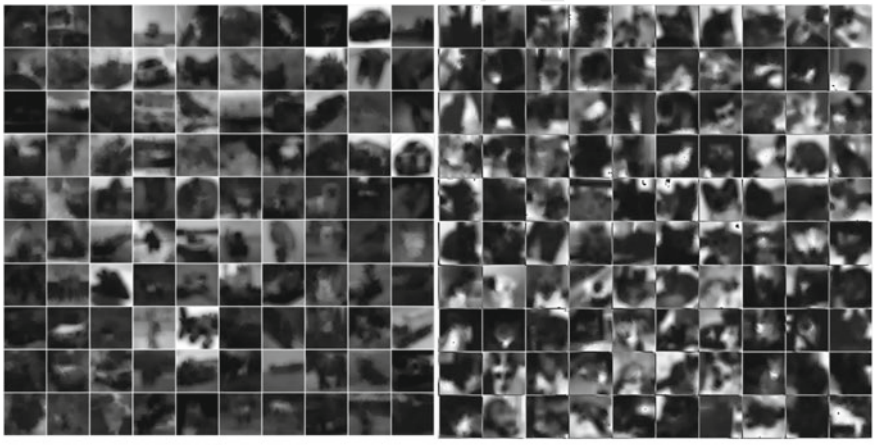


Fig. 13 Transmission map estimation. **a** Transmission map over 100 images from CIFAR-10; **b** Transmission map over 100 images from Dogs and Cats dataset

References

1. Cohen, T., Welling, M.: Group equivariant convolutional networks. In: International Conference on Machine Learning, pp. 2990–2999 (2016)
2. Cohen, T., Geiger, M., Köhler, J., Welling, M.: Convolutional networks for spherical signals. arXiv preprint arXiv:1709.04893 (2017)
3. Anselmi, F., Leibo, J.Z., Rosasco, L., Mutch, J., Tacchetti, A., Poggio, T.: Unsupervised learning of invariant representations. Theor. Comput. Sci. **633**, 112–121 (2016)
4. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In: International Conference on Document Analysis and Recognition (ICDAR), p. 958. IEEE Computer Society (2003)
5. Weinstein, M., Breiding, P., Sturmfels, B., Kališnik Verovšek, S.: Learning algebraic varieties from samples. Revista Matemática Complutense **31**, 545–593 (2018)
6. Bronstein, M.M., Bruna, J., LeCun, Y., Szlam, A., Vandergheynst, P.: Geometric deep learning: going beyond Euclidean data. IEEE Sig. Process. Mag. **34**(4), 18–42 (2017)
7. Mallat, S.: Understanding deep convolutional networks. Philos. Trans. Roy. Soc. A **374**(2065), 20150203 (2016)
8. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. **35**(8), 1872–1886 (2013)
9. Wang, X., Jin, X., Xu, G., Xu, X.: A multi-scale decomposition based haze removal algorithm. In: International Conference on Remote Sensing, Environment and Transportation Engineering (RSETE), Nanjing, China, vol. 2, pp. 1–4, June 2012
10. Sifre, L., Mallat, S.: Rotation, scaling and deformation invariant scattering for texture discrimination. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1233–1240 (2013)
11. Hestenes, D., Sobczyk, G.: Clifford Algebra to Geometric Calculus: A Unified Language for Mathematics and Physics, vol. 5. Springer, Dordrecht (2012)
12. Xambó-Descamps, S.: Real Spinorial Groups–A Short Mathematical Introduction. SBMA/Springerbrief. Springer, Heidelberg (2018)
13. Lavor, C., Xambó-Descamps, S., Zaplana, I.: A Geometric Algebra Invitation to Space-Time Physics Robotics and Molecular Geometry. SBMA/Springerbrief. Springer, Heidelberg (2018)
14. Felsberg, M.: Low-Level Image Processing with the Structure Multivector, vol. 203. Inst. für Informatik und Praktische Mathematik (2002)
15. Mitrea, M.: Clifford Wavelets, Singular Integrals, and Hardy Spaces. Springer, Heidelberg (2006)
16. Chan, W.L., Choi, H., Baraniuk, R.: Quaternion wavelets for image analysis and processing. In: 2004 International Conference on Image Processing, ICIP 2004, vol. 5, pp. 3057–3060, IEEE (2004)
17. Hitzer, E., Sangwine, S.J.: Quaternion and Clifford Fourier Transforms and Wavelets. Springer, Heidelberg (2013)
18. Felsberg, M., Sommer, G.: The monogenic signal. IEEE Trans. Sig. Process. **49**(12), 3136–3144 (2001)
19. Moya-Sánchez, E.U., Bayro-Corrochano, E.: Quaternion atomic function wavelet for applications in image processing. In: Iberoamerican Congress on Pattern Recognition, pp. 346–353. Springer (2010)
20. Bayro-Corrochano, E., Vazquez-Santacruz, E., Moya-Sanchez, E., Castillo-Muñis, E.: Geometric bioinspired networks for recognition of 2-D and 3-D low-level structures and transformations. IEEE Trans. Neural Netw. Learn. Syst. **27**(10), 2020–2034 (2016)
21. Isokawa, T., Matsui, N., Nishimura, H.: Quaternionic neural networks: fundamental properties and applications. In: Complex-Valued Neural Networks: Utilizing High-Dimensional Parameters, pp. 411–439. IGI Global (2009)
22. Buchholz, S., Sommer, G.: Quaternionic spinor MLP. In: ESANN 2000 Proceedings, D-Facto, European Symposium on Artificial Neural Networks, Bruges, Belgium, pp. 377–382, 26–28 April 2000 (2000)

23. Kominami, Y., Ogawa, H., Murase, K.: Convolutional neural networks with multi-valued neurons. In: 2017 International Joint Conference on Neural Networks (IJCNN), pp. 2673–2678. IEEE (2017)
24. Parcollet, T., Zhang, Y., Morchid, M., Trabelsi, C., Linarès, G., De Mori, R., Bengio, Y.: Quaternion convolutional neural networks for end-to-end automatic speech recognition. arXiv preprint arXiv:1806.07789 (2018)
25. Zhu, X., Xu, Y., Xu, H., Chen, C.: Quaternion convolutional neural networks. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 631–647 (2018)
26. Gaudet, C., Maida, A.: Deep quaternion networks. arXiv preprint arXiv:1712.04604 (2017)
27. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Technical report, Citeseer (2009)
28. Elson, J., Douceur, J.R., Howell, J., Saul, J.: Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In: ACM Conference on Computer and Communications Security, vol. 7, pp. 366–374 (2007)
29. Wang, Z., Simoncelli, E.P.: Local phase coherence and the perception of blur. In: Advances in Neural Information Processing Systems, pp. 1435–1442 (2004)
30. Sierra-Vázquez, V., Serrano-Pedraza, I.: Application of Riesz transforms to the isotropic AM-PM decomposition of geometrical-optical illusion images. J. Opt. Soc. Am. A **27**, 781–796 (2010)
31. Moya-Sánchez, E.U., Vázquez-Santacruz, E.: A geometric bio-inspired model for recognition of low-level structures. In: International Conference on Artificial Neural Networks, pp. 429–436. Springer (2011)
32. Tewari, A.: Image blending using local phase. Master of Science in Informatics at Grenoble, Université Joseph Fourier (2015)
33. Wadhwa, N., Rubinstein, M., Durand, F., Freeman, W.T.: Riesz pyramids for fast phase-based video magnification. In: 2014 IEEE International Conference on Computational Photography (ICCP), pp. 1–10. IEEE (2014)
34. Unser, M., Sage, D., Van De Ville, D.: Multiresolution monogenic signal analysis using the Riesz-Laplace wavelet transform. IEEE Trans. Image Process. **18**(11), 2402–2418 (2009)
35. Felsberg, M., Sommer, G.: A new extension of linear signal processing for estimating local properties and detecting features. In: Mustererkennung 2000, pp. 195–202. Springer (2000)
36. González, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB, 2nd edn. Tata McGraw-Hill (2010). MATLAB examples
37. Kovesi, P.D.: MATLAB and Octave functions for computer vision and image processing (2018). https://www.peterkovesi.com/
38. Granlund, G.H., Knutsson, H.: Signal Processing for Computer Vision. Springer, Dordrecht (2013)
39. Agoston, M.K.: Computer Graphics and Geometric Modeling, vol. 1. Springer, London (2005)
40. McCartney, E.J., Hall, F.F.: Optics of the atmosphere: scattering by molecules and particles. Phys. Today **30**(5), 76–77 (1977)
41. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European Conference on Computer Vision, pp. 630–645. Springer (2016)
42. Boukerroui, D., Noble, J.A., Brady, M.: On the choice of band-pass quadrature filters. J. Math. Imaging Vis. **21**(1–2), 53–80 (2004)