A Quaternion Monogenic Layer Resilient to Large Brightness Changes in Image Classification

Anonymous Author(s)

Affiliation
Address
email

1 Broader Impact

We share the common concern of non-ethical applications of research in machine learning. There is also the risk of possible involuntary collateral effects produced by unexpected responses of layers such as M6 in unfamiliar scenarios. Nevertheless, these concerns do not yet bare on our work. Because of its academic nature, as disclosed by the simple nature of the datasets used, we cannot think about real applications as yet. These will certainly require a bigger research effort. Real world applications will be based on more involved CNNs architectures, as they ought to be capable of performing operations such as image segmentation, object detection, face recognition, etc. in real time and under variable brightness conditions. If we are to consider benefits of our work, the most immediate is for researchers that are in the look for increased resilience of their models in front of brightness variations. Beyond that, we expect that it can be useful for dealing with scenarios in which unpredictable large brightness changes occur.

There are two limitations of our work that have to be mentioned. One is that we found our setup works poorly for \( \alpha > 0.6(255) \). The other concern is that our implementation is not optimized, which is the reason why the timings reported for the M6 architectures were greater than those for the C architectures (about 29% for training and 20% for testing).

2 Background

Currently, there is no conventional definition for brightness. In fact, image-processing tools employ several different brightness measurements [1]. Brightness refers to the overall lightness or darkness of the image [2]. In image processing and computer vision, changing brightness of an image is a commonly used point transformation (affecting every pixel in an image). In this transformation, the value of each pixel is increased by a constant. For a one channel image \( I = I(x,y) \in \mathbb{R} \) (where \( x, y \in U \), \( U \) a region of \( \mathbb{R}^2 \)) the relation

\[
I_B(x,y) = \min(I(x,y) + \alpha, 255),
\]

where \( \alpha > 0 \) is a constant, defines an image \( I_B \) that is brighter than \( I \), the more so the higher the value of \( \alpha \). In Figure [1] we can see an original image \( B_0 \) and brighter versions \( B_i \) \((i = 1, 2, 3)\) corresponding to three values of \( \alpha \). In addition, we display the histogram of the four images. Notice that the contrast also changes and that pixels with \( I(x,y) + \alpha \geq 255 \) become saturated.

We have changed the brightness of all datasets using equation [1] implemented in Tensorflow 2.1. Figure [2] displays the point transformation from \( I(x,y) \) to \( I_B(x,y) \). We use degradation labels \( B_i, i = 0, 1, 2, 3, \) where \( B_0 \) corresponds to \( \alpha = 0 \) (solid line in Figure [2]), and \( B_1, B_2, B_3 \) to \( \alpha = 0.3(255), 0.4(255) \) and \( 0.5(255) \).

We define 1D (resp. 2D) multivectorial signals as $C^1$ maps $U \rightarrow \mathcal{G}$ from an interval $U \subset \mathbb{R}$ (a region $U \subset \mathbb{R}^2$) into a geometric algebra $\mathcal{G}$ (see [3]). For $\mathcal{G} = \mathbb{R}$ ($\mathcal{G} = \mathbb{C}$, $\mathcal{G} = \mathbb{H}$) 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 maps 2D scalar signals to 2D quaternionic signals. Among the signals obtained in this way, our interest lies in the (quaternionic) monogenic signals (see [4] for details). We use a band-pass monogenic signal $I_M = I_M(x,y) \in \mathbb{H}$ associated to an image $I = I(x,y) \in \mathbb{R}$ (where $x, y \in U$, $U$ a region of $\mathbb{R}^2$). The definition of the band-pass $I_M$ is as follows:

$$I_M = I' + I_R, \quad I_R = iI_1 + jI_2,$$

(2)

where, $I' = g \ast I$, $\ast$ is the convolution operator, $g = g(x,y)$ is a radial (isotropic) bandpass (Log-Gabor function), the signals $I_1$ and $I_2$ are the Riesz transforms (with quadrature filters) of $I'$ in the $x$ and $y$ directions [4]. Note that $I_M \in \langle 1, i, j \rangle \subset \mathbb{H}$. Rewriting equations in Fourier domain we have:

$$I_M = \mathcal{F}^{-1}(J' + J_R), \quad J_R = iJ_1 + jJ_2,$$

(3)
where

\[ J' = J \cdot G, \quad J_1 = J \cdot H_1 \cdot G, \quad J_2 = J \cdot H_2 \cdot G, \]

\[ J[u_1, u_2] = \sum_{m_1} \sum_{m_2} I[m_1, m_2] e^{-2\pi(i(u_1 m_1 + u_2 m_2))} \]  \hspace{1cm} (4)

\[ H_1(u_1, u_2) = \frac{u_1}{\sqrt{u_1^2 + u_2^2}}, \]  \hspace{1cm} (5)

\[ H_2(u_1, u_2) = \frac{u_2}{\sqrt{u_1^2 + u_2^2}}, \]  \hspace{1cm} (6)

\[ G(u_1, u_2) = \exp \left( - \frac{\log \left( \sqrt{u_1^2 + u_2^2} / \omega_0 \right)^2}{2 \log(\sigma)^2} \right), \]  \hspace{1cm} (7)

\[ \omega_0^s = \frac{1}{\min_w f^{s-1}} \]  \hspace{1cm} (8)

where \( u_1, u_2 \) are frequency components, \( J \) is 2D Fourier transform \( \mathcal{F} \) of \( I \), \( \min_w \) is the minimum wavelength, \( f \) is a scale factor, \( s = 1, 2, \ldots, n_s \) is the current scale.

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 \( I_M(x, y) \) \[4\]. Notice that we have

\[ |I_M| = \sqrt{I_1^2 + I_2^2}, \]  \hspace{1cm} (9)

similarly \( |I_R| = \sqrt{I_1^2 + I_2^2} \). The **local phase** \( I_\phi \) and the **local orientation** \( I_\theta \) associated to \( I \) are defined, following \[4\], by the relations

\[ I_\phi = \text{atan2} \left( \frac{I'}{|I_R|} \right), \]  \hspace{1cm} (10)

\[ I_\theta = \text{atan} \left( \frac{-I_2}{I_1} \right), \]  \hspace{1cm} (11)

where the quotients of signals are taken point-wise. For the geometric interpretation of these signals see Figure [5]

![Figure 3: Geometry of the monogenic signal.](image)

### 3.2 Monogenic layer

The monogenic layer M6 (cf. \[5\]) is best described by the scheme in Figure 4 where 1 in the HSV representation is a ones matrix of \([m, n] \). The Normalization is defined as

\[ \text{Normalization}(I) = \frac{I(x, y) - \min(I(x, y))}{\max(I(x, y)) - \min(I(x, y))}, \]  \hspace{1cm} (12)

The \( (HSV2RGB) \) transforms an HSV image into an RGB image according to the standard color-naming conventions (see page 304 of \[6\]). See Figure 5 for an illustration of the M6 components
of a simple gray image. Remark that $RGB_\phi$ enhances lines and edges and $RGB_\theta$ enhances the orientation components all over the image. Figure 6 illustrates the six feature maps from image example. In Table 1 we present the main characteristics of a conventional CNN layer $C$ and the M6 layer. Note that the M6 is defined in frequency domain as a result we only have 4 parameters in the layer.

Figure 4: $RGB_\theta$ and $RGB_\phi$ are the outputs of the M6 Layer.

The implementation of M6 has been coded using Tensorflow 2.1 (TF) and Keras [7,8].

Appendix 1. Quaternion algebra

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

$$\mathbb{H} = \mathbb{R}1 \oplus \mathbb{R}i \oplus \mathbb{R}j \oplus \mathbb{R}k$$

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

$$i^2 = j^2 = k^2 = ijk = -1.$$  \(\text{(14)}\)

As it is easily seen, these relations imply that

$$ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j.$$  \(\text{(15)}\)

Figure 5: Feature maps of M6 from a circle image input $I(x, y)$.

Figure 6: (A) $RGB$ input image. (B) $RGB_\theta$ and (C) $RGB_\phi$ are the output feature maps of the M6 layer.

The implementation of M6 has been coded using Tensorflow 2.1 (TF) and Keras [7,8].

Appendix 1. Quaternion algebra

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

$$\mathbb{H} = \mathbb{R}1 \oplus \mathbb{R}i \oplus \mathbb{R}j \oplus \mathbb{R}k$$

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

$$i^2 = j^2 = k^2 = ijk = -1.$$  \(\text{(14)}\)

As it is easily seen, these relations imply that

$$ij = -ji = k, \quad jk = -kj = i, \quad ki = -ik = j.$$  \(\text{(15)}\)

Figure 5: Feature maps of M6 from a circle image input $I(x, y)$.

Figure 6: (A) $RGB$ input image. (B) $RGB_\theta$ and (C) $RGB_\phi$ are the output feature maps of the M6 layer.
The elements of $\mathbf{H}$ are named quaternions, and $i, j, k$, quaternionic units. By definition, a quaternion $q$ can be written in a unique way in the form

$$ q = a + bi + cj + dk, \quad a, b, c, d \in \mathbb{R}. $$

(16)

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

$$ \bar{q} = a - (bi + cj + dk). $$

(17)

Note that $(q + \bar{q})/2 = a$, which is called the real part or scalar part of $q$, and $(q - \bar{q})/2 = a = bi + cj + dk$, the vector part of $q$. Since the conjugates of $i, j, k$ are $-i, -j, -k$, the relations $i^2 = j^2 = k^2 = -1$, $ij = k$, $jk = i$, $ki = j$, $ij = -ji$, $jk = -kj$, $ki = -ik$, and $i^2 = j^2 = k^2 = -1$ 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

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

(18)

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

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

(19)

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

### References


### Table 1: Comparison of the main characteristics of a standard convolutional layer $C$ and the M6 layer.

<table>
<thead>
<tr>
<th>Characteristics/Name</th>
<th>$C$</th>
<th>M6</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Parameters</strong></td>
<td>$[k_1 \times k_2 \times l]$</td>
<td>4</td>
</tr>
<tr>
<td>Convolution domain</td>
<td>Space</td>
<td>Frequency (Fourier)</td>
</tr>
<tr>
<td>Output shape (with input size $[m, n]$, and padding)</td>
<td>$[m, n, l]$</td>
<td>$[m, n, 6]$</td>
</tr>
<tr>
<td>Output domain</td>
<td>Space</td>
<td>Space</td>
</tr>
<tr>
<td>Nonlinear function</td>
<td>ReLU</td>
<td>arctan, arcsin</td>
</tr>
<tr>
<td>Layer position</td>
<td>Any</td>
<td>First hidden</td>
</tr>
<tr>
<td>Trainable</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>