Quaternion Convolutional Neural Network for Color Image Classification and Forensics

QILIN YIN JINWEI WANG XIANGYANG LUO JIANGTAO ZHAI SUNIL KR. JHA AND YUN-QING SHI

March 1, 2022

IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 4, 2019, accepted January 24, 2019, date of publication February 1, 2019, date of current version February 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2897000

Quaternion Convolutional Neural Network for Color Image Classification and Forensics

QILIN YIN[1], JINWEI WANG[1,2,3], (Member, IEEE), XIANGYANG LUO[3], (Member, IEEE), JIANGTAO ZHAI[2], SUNIL KR. JHA[2], AND YUN-QING SHI[4], (Fellow, IEEE)

[1]Department of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology, Nanjing University of Information Science and Technology, Nanjing 210044, China
[3]State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou 450001, China
[4]Department of Electrical Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102, USA

Corresponding author: Jinwei Wang (wjwei_2004@163.com)

This work was supported in part by the Natural Science Foundation of China under Grant 61772281, Grant U1636219, Grant 61702235, Grant 61502241, Grant 61272421, Grant 61232016, Grant 61402235, and Grant 61572258, in part by the National Key R&D Program of China under Grant 2016YFB0801303 and Grant 2016QY 01W0105, in part by the Plan for Scientific Talent of Henan Province under Grant 2018JR0018, in part by the Natural Science Foundation of Jiangsu Province, China, under Grant BK20141006, in part by the Natural Science Foundation of the Universities in Jiangsu Province under Grant 14KJB520024, in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions Fund, and in part by the Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology fund.

ABSTRACT The convolutional neural network is widely popular for solving the problems of color image feature extraction. However, in the general network, the interrelationship of the color image channels is neglected. Therefore, a novel quaternion convolutional neural network (QCNN) is proposed in this paper, which always treats color triples as a whole to avoid information loss. The original quaternion convolution

Table of Content

Introduction

A traditional convolutional neural network consists of one or several convolutional layers, followed by some fully-connected layers of neurons. Each convolution block usually produces feature maps by four steps, e.g.,**convolution**, **batch normalization**, **non-linear activation**, and **pooling**.

In the last two years, attention model has been widely used in various types of deep learning tasks such as natural language processing, computer vision and speech recognition. From the naming of attention model, it is obvious that it borrows from the human vision attention mechanism. Human vision rapidly focuses on the target area by scanning the global image, and then invests more attention to obtain more detailed information of the target and suppress useless information. Just like this, through the attention model, the neural network can autonomously and purposefully select key features and discard redundant features.

Introduction

The traditional CNN framework is becoming more and more mature, but it still has some drawbacks, like when dealing with the color images, general CNNs just treat the RGB three channels as three unrelated feature maps. Although during the convolution process, per convolution kernel sum up the convolution result of different channels as a single output, this still neglects the interrelationship of the RGB three color channels, resulting in the loss of feature information.

Introduction

We propose a quaternion convolutional neural network (QCNN) model. a color image can be represented as a quaternion matrix, each element of which is a pure quaternion consisting of 3-tuple (RGB or LMS). The advantage of this method is that a color can be processed as a hypercomplex number, rather than three separate channels, reserving the interrelationship information between the color channels mostly. So, a color image is circulated and processed in the form of a quaternion matrix in the QCNN. Compared with the traditional CNNs, QCNN not only preserves the vertical relationship between the features, but also maintains the horizontal relationship of the color channels.

Introduction

The essence of attention mechanism is to reuse feature maps, give new weight to feature maps and mark out key features. Regardless of the real domain or hypercomplex, this idea is universal. We try to incorporate an attention mechanism into the QCNN model to improve the proposed model performance further. The following experimental results show that our model can outperform the traditional CNN model and the other QCNN model, which has only recently been proposed and is called Pure QCNN by us below.

Quatrnion Algebra

A quaternion can be defined as:

$$q = a + bi + cj + dk$$

where $a, b, c, d \in \mathbb{R}$, and the imaginary units i, j, k obey the quaternion rules that $i^2 = j^2 = k^2 = ijk = -1$.If $a = 0$, we can call q is a pure quaternion.

Quatrnion Algebra

- Addition:

$$q_1 + q_2 = (a_1 + a_2) + (b_1 + b_2)i + (c_1 + c_2)j + (d_1 + d_2)k$$

- Norm:

$$|q| = \sqrt{qq^*} = \sqrt{a^2 + b^2 + c^+ d^2}$$

- Conjugation:

$$q^* = a - bi - cj - dk$$

- Quaternion multiplication:

$$\begin{aligned} q_1 q_2 =& (a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2) \\ &+ (a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2)i \\ &+ (a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2)j \\ &+ (a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2)k \end{aligned}$$

Let $W = A + Bi + Cj + Dk$ be a quaternion convolution kernel matrix, and $X = a + bi + cj + dk$ the quaternion input matrix. When a color image is involved, the real part a is set to 0, and the input matrix becomes the pure quaternion matrix.Similar to the traditional convolution, the quaternion-valued convolution $W \otimes X$ can be defined as follows:

$$
\begin{aligned}
W \otimes X =& (Aa - Bb - Cc - Dd) \\
& + (Ab + Ba + Cd - Dc)i \\
& + (Ac - Bd + Ca + Db)j \\
& + (Ad + Bc - Cb + Da)k
\end{aligned} \tag{1}
$$

The quaternion mean and variance have been defined as follows.

$$\begin{aligned} QE(x) =& \frac{1}{T}\sum_{i=1}^{T} q_0 + q_1 i + q_2 j + q_3 k \\ =& \overline{q_0} + \overline{q_1} i + \overline{q_2} j + \overline{q_3} k \\ QV(x) =& \frac{1}{T}\sum_{i=1}^{T} (x - QE(x))(x - QE(x))^* \\ =& \frac{1}{T}\sum_{i=1}^{T} (\Delta q_0^2 + \Delta q_1^2 + \Delta q_2^2 + \Delta q_3^2) \end{aligned} \tag{2}$$

where $x = q_0 + q_1 i + q_2 j + q_3 k, \Delta q_i = q_i - \overline{q_i}$

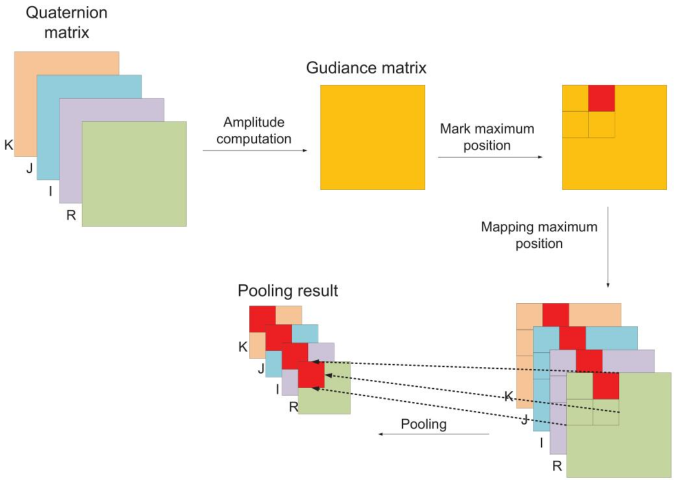
The quaternion batch normalization $QBN(x_i)$ can be defined as:

$$
\begin{aligned}
QBN(x_i) =& \gamma(\frac{x_i - QE(x)}{\sqrt{QV(x) + \epsilon}}) + \beta \\
=& \gamma(\frac{q_0^i - \overline{q_0}}{\sqrt{QV(X) + \epsilon}} + \frac{(q_1^i - \overline{q_1})i}{\sqrt{QV(X) + \epsilon}} \\
& + \frac{(q_2^i - \overline{q_2})i}{\sqrt{QV(X) + \epsilon}} + \frac{(q_3^i - \overline{q_3})i}{\sqrt{QV(X) + \epsilon}}) \\
& + (\beta_0 + \beta_1 i + \beta_2 j + \beta_3 k)
\end{aligned}
\tag{3}
$$

Here, γ is a scalar that initializes to 1, representing stretch scale, β is a quaternion that initializes to 0, representing shift scale, ϵ is a non-zero minimum. γ and β are trainable parameters that participate in network weight updates.

There are many types of pooling layers, e.g., Max-pooling and Mean-pooling, in the real-valued neural network. They all can be extended to hypercomplex domain. For the Mean-pooling operation, pooling the real part and three imaginary parts of the quaternion matrix separately will not affect final pooling result.

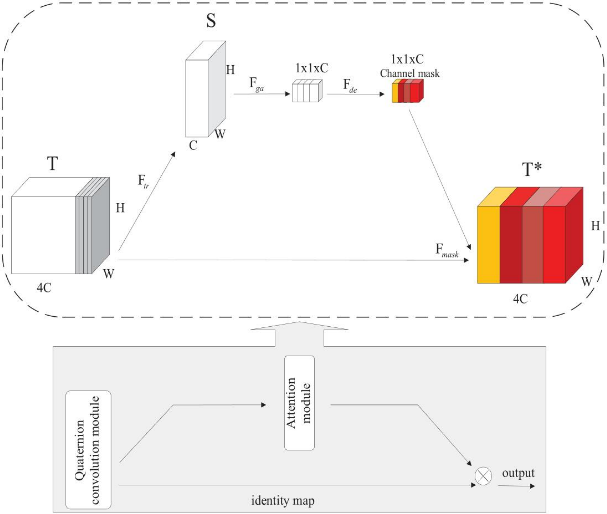
However, in terms of Max-pooling, if we pool each part individually, this will create a data mess. Because we cannot make sure the position of the maximum of each part is each part is corresponding, we can use some algorithms to get the guidance matrix for the quaternion matrix. Then, according to the max pooling result of the guidance matrix, the four parts of the quaternion matrix can be simplified. Combining the four parts, the max pooling of the entire quaternion matrix is completed finally.



Quaternion matrix
K
J
I
R
Amplitude computation
Gudiance matrix
Mark maximum position
Mapping maximum position
Pooling result
K
J
I
R
Pooling
K
J
I
R

Components of Quaternion CNN

Quaternion Attention Module



Let $T \in H \times W \times 4C$ be the input of the quaternion attention module, the number of channel $4C$ shows the uniqueness of the proposed quaternion network too. Because of the characteristic, we need to modify the attention module in the general network accordingly to accommodate the data format of a quaternion. Firstly, T is transformed by an operation F_{tr} to generate guidance data $S \in H \times W \times C$. When it comes to compressing quaternion data, the transformation features of quaternion provide us different choices just like what we do with the quaternion pooling operation. Secondly, S is pooled by a global average pooling $F_{ga}(.)$ to estimate statistical characteristics of each channel distribution. This process can be formulated as

$$z = \frac{1}{H \times W} \sum_{j=1}^{H} \sum_{i=1}^{W} S(i,j), z \in 1 \times 1 \times C \tag{4}$$

Thirdly, z only represents the channel priority distribution for a mini-batch sample, which is not applicable to the entire training set, let alone the test set. Therefore, multiple full connection operations $F_{de}(.)$ should be added to the quaternion module to train a generalized channel weight mask using a correlation between channels. This process can be formulated as

$$\widetilde{z} = F_{de}(W, z) = \theta(W_2\sigma(W_1 z)), W_1 \in \frac{C}{r} \times C, W_2 \in C \times \frac{C}{r} \tag{5}$$

$W_1 z$ is a full connection operation, σ is a activation operation that is adopted in Relu activation function, $W_2\sigma(W_1 z)$ is another a full connection operation, θ is also a activation function, but Sigmoid activation function is chosen. r is a hyperparameter, it determines the proportion of data compression to reduce the amount of computation.

Finally, channel mask $\widetilde{z}$ is combined with T by operation $F_{mask}(.)$ to highlight beneficial features and inhibit unbeneficial ones.

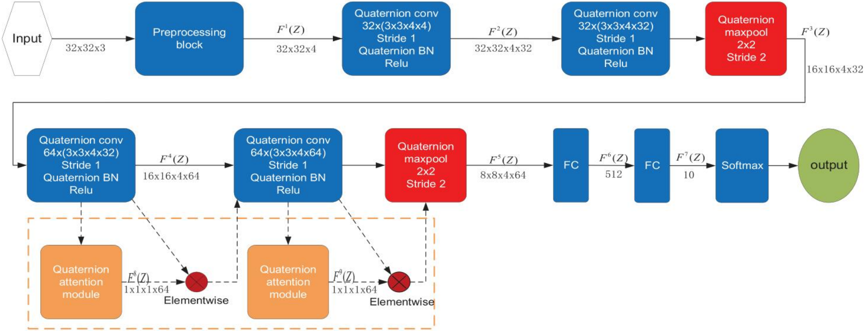
$$T^* = F_{mask}(\widetilde{z}, T) = \widetilde{z}T \tag{6}$$

$$\mathbb{F}(q) = \mathbb{F}(a) + \mathbb{F}(b)i + \mathbb{F}(c)j + \mathbb{F}(d)k \tag{7}$$

where $\mathbb{F}$ corresponding to any standard activation function.

Experiment

Architecture



Experiment

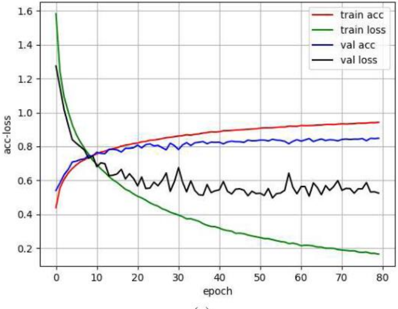
Architecture

In order to test the performance of the QCNN component module, we built a basic QCNN. It contains two convolution blocks, two max-pooling layers, and end with two fully-connected layers. Each convolution block is composed of two quaternion-valued convolution layers, each of which contains convolution, batch normalization, and activation operations. ReLU is used as a standard activation function. Different from traditional convolutional neural networks, each convolution kernel and feature map in the proposed model is a quaternion matrix with a size of $width \times height \times 4$. Note that the quaternion attention module shown in the dotted orange box is not an essential component of the basic QCNN. It is just a method to improve network performance.

Model	Data set	Test accuracy
QCNN (with traditional batch normalization)	Cifar-10	0.8309
QCNN (with quaternion batch normalization)	Cifar-10	**0.8415**

Experiment

Impact of the Quaternion Attention Module



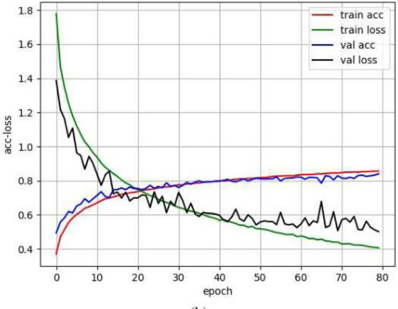


FIGURE 5. The loss and accuracy of the QCNN and the attention based QCNN during the training and testing. (a) QCNN. (b) Attention_based QCNN.

Model	Data set	Test accuracy
QCNN	Cifar-10	0.8415
attention based QCNN	Cifar-10	**0.8537**

TABLE 3. Experiment results in classification tasks.

Model	Data set	Test accuracy
Real network	Cifar-10	0.7546
Pure QCNN	Cifar-10	0.7778
QCNN(proposed)	Cifar-10	**0.8415**
attention based QCNN(proposed)	Cifar-10	**0.8537**

TABLE 4. Experiment results of two methods on UCID.

QF	Huang's	Proposed
90	0.8956	**0.9917**
80	0.8380	**0.9846**
70	0.7246	**0.7783**