Mathematical Biosciences and Engineering

MBE, 16 (6): 6907–6922.
DOI: 10.3934/mbe.2019346
Received: 04 March 2019
Accepted: 22 July 2019
Published: 29 July 2019

http://www.aimspress.com/journal/MBE

Research article

Color image splicing localization algorithm by quaternion fully convolutional networks and superpixel-enhanced pairwise conditional random field

Beijing Chen [1,2,3,*], Ye Gao [1], Lingzheng Xu [4], Xiaopeng Hong [5], Yuhui Zheng [1,2] and Yun-Qing Shi [1,6]

[1] Jiangsu Engineering Center of Network Monitoring, School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing 210044, China
[2] Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science & Technology, Nanjing 210044, China
[3] Key Laboratory of Computer Network Technology of Jiangsu Province, Southeast University, Nanjing 210096, China
[4] College of Computer Science, Sichuan University, Chengdu 610065, China
[5] Center for Machine Vision and Signal Analysis, University of Oulu, Oulu FI-90014, Finland
[6] Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark 07102, USA

* **Correspondence:** Email: nbutimage@126.com.

Abstract: Recently, fully convolutional network (FCN) has been successfully used to locate spliced regions in synthesized images. However, all the existing FCN-based algorithms use real-valued FCN to process each channel separately. As a consequence, they fail to capture the inherent correlation between color channels and the integrity of three channels. So, in this paper, quaternion fully convolutional network (QFCN) is proposed to generalize FCN to quaternion domain by replacing real-valued conventional blocks in FCN with quaternion conventional blocks. In addition, a new color image splicing localization algorithm is proposed by combining QFCNs and superpixel (SP)-enhanced pairwise conditional random field (CRF). QFCNs consider three different versions (QFCN32, QFCN16, and QFCN8) with different up-sampling layers. The SP-enhanced pairwise CRF is used to refine the results of QFCNs. Experimental results on three publicly available datasets demonstrate that the proposed algorithm outperforms the existing algorithms including some conventional algorithms and some deep learning-based algorithms.

Keywords: splicing localization; splicing detection; quaternion; fully convolutional network; conditional random field

1. Introduction

The popularity of cameras and the development of the Internet make digital images widely integrated into people's daily life. However, with the rapid development of image editing software tools, it is getting easier to edit and tamper images without leaving any obvious visual traces [1,2]. Since images can be regarded as evidence or proof in the fields of news, academic, politics, crime investigation and insurance claims investigation, malicious forged images may bring many considerable adverse problems to the society. For example, Los Angeles Times published a synthesized photograph of a British soldier urging Iraqi civilians to seek cover in March 2003, which made a 20-year veteran news photographer Brian Walski fired; a forged image of John Kerry and Jane Fonda's speeches in an anti-Vietnamese war protests was circulated during the 2004 US presidential election, which brought irreparable adverse effects to John Kerry. Image splicing is one of the most common types of image forgery. It creates synthesized images by cropping and pasting regions from one or more source images carefully. Therefore, it is important to develop reliable algorithms for image splicing localization. Image splicing localization endeavors to locate accurate regions. It is more difficult than splicing detection that simply distinguishes whether a given image is authentic or synthesized. In addition, image splicing localization is different from saliency detection or object detection. Image splicing localization does not pay much attention to image content as saliency detection and object detection do. Instead, it focuses on detecting traces left by tampering operations, such as strong contrast difference, unnatural tampered boundary, sensor noise, compression artifacts, and so on.

Over the years, a variety of algorithms for image splicing localization have been proposed. These algorithms rely on an assumption that spliced regions essentially carry information that is in some significant aspects different from the rest of the image. There exist some clues that can be used to locate the splicing region, such as sensor noise [3–5], color filter array (CFA) artifacts [6–8], misaligned JPEG blocks [9–12], compression quantization artifacts [13] and resampling artifacts [14,15]. However, all of these conventional algorithms are based on hand-crafted features. It limits these algorithms to deal with one or some certain types of forgery.

So, some recent works have moved away from using prior assumptions and applied deep-learning for splicing localization. Li et al. [16] proposed two detectors (a statistical feature based detector and a copy-move forgery detector) to analyze the input images and merged detection results to generate location maps. Cozzolino et al. [17] showed that residual-based descriptor could be regarded as a simple constrained convolutional neural network (CNN) which can conduct forgery detection and localization. Liu et al. [18] first utilized CNNs for color patches of different scales to generate real-valued tampering possibility maps and then fused the generated maps to get the localization map. However, the above-mentioned algorithms are block-based algorithms, accordingly, they only provide block-level accuracy. In order to obtain pixel-level accuracy, some methods have been proposed. Bappy et al. [19] utilized a long short-term memory (LSTM) based multi-task network to learn the labels of small image patches and used pixel-wise segmentation to determine each pixel forged or not. Shi et al. [20] designed a dual-domain-based convolutional neural network (D-CNN) for different kinds of inputs (spatial domain-based and frequency domain-based) and

applied two post-processing operations to finalize pixel-wise forged region localization. Salloum et al. [21] used multi-task fully convolutional network (MFCN) to learn surface label and edge of the spliced regions respectively. Liu et al. [22] utilized three different fully convolutional networks (FCNs) to locate spliced regions respectively and then fused the predictions of three FCNs by CRF to obtain the final location map. Chen et al. [23] proposed an improved splicing localization algorithm to make the work by Liu et al. [22] be an end-to-end learning system. They also utilized region proposal network (RPN) to enhance learning ability of object areas because forgery usually happens in the object areas. Bappy et al. [24] proposed a two-stream network which exploits the features in both frequency domain and spatial domain to locate forged regions by incorporating encoder and LSTM network. However, for color forged images, all these deep learning-based methods use real-valued CNNs to process each channel separately [25]. As a consequence, they fail to capture the inherent correlation between color channels and the integrity of three channels [26].

The quaternion is an extension of complex number. During the past two decades, it has been regarded as a tool of color images processing by encoding their three channels into the imaginary parts of the quaternion representation (QR) [2,25–27]. The two main advantages of QR are that: (a) it helps capture the inherent correlation between color channels; (b) it treats a color image as a vector field. So, using the QR and quaternion algebra, many classical tools developed for gray-scale image have been successfully extended to color image processing, such as Fourier transform [26,27], neural networks [28], principal component analysis [29], kernel quaternion principal component analysis [30], fractional Fourier transform [31], fractional cosine transform [2], and discrete fractional random transform [32], etc. Recently, CNN as a powerful feature representation method has achieved fine performance in almost all vision tasks [33–37]. So, some researchers also investigated the extensions of the CNN in quaternion domain and proposed quaternion CNN (QCNN) model [25,38,39]. QCNN model has been shown to achieve better results than the traditional CNN model in both of color image classification task [38] and color image segmentation task [39].

In [21–23], FCN-based algorithms have been shown to outperform some deep learning-based algorithms [16–20] and some conventional algorithms [3–15] in image splicing localization. So, in this paper, we propose quaternion fully convolutional neural network (QFCN) to generalize FCN to quaternion domain. Then, QFCN is used for color image splicing localization by combining with superpixel (SP)-enhanced pairwise CRF. SP-enhanced pairwise CRF is used to refine the results obtained from QFCN. Similar to [22,23], three different versions of QFCNs (QFCN32, QFCN16, and QFCN8) with different up-sampling layers are considered together.

This paper is organized as follows. In Section 2, we recall quaternion color representation and some layers in QFCN. Section 3 describes the proposed color image splicing localization algorithm. Experimental results and analysis are provided in Section 4. Finally, Section 5 concludes this paper.

2. Some preliminaries

This section recalls QR and some layers in QFCN.

2.1. Quaternion number and quaternion color representation

Quaternion numbers are the generalization of complex numbers. A quaternion number has one real part and three imaginary parts as

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

where a, b, c, and d are four real numbers, and $\boldsymbol{i}$, $\boldsymbol{j}$, $\boldsymbol{k}$ are three imaginary units obeying the following rules

$$\begin{aligned} &\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = -1, \quad \boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \\ &\boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}, \ \ \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}. \end{aligned} \tag{2}$$

When the real part $a = 0$, q is called a pure quaternion. The conjugate and modulus of a quaternion number are respectively defined as

$$q^* = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}\,, \tag{3}$$

$$\|q\| = \sqrt{a^2 + b^2 + c^2 + d^2}\,. \tag{4}$$

Let $f(u, v)$ be an RGB image function, each pixel can be represented as a pure quaternion by the QR

$$f(u,v) = f_R(u,v)\boldsymbol{i} + f_G(u,v)\boldsymbol{j} + f_B(u,v)\boldsymbol{k}\,, \tag{5}$$

where $f_R(u, v)$, $f_G(u, v)$, $f_B(u, v)$ are respectively the red, green and blue components of the pixel (u, v).

2.2. Quaternion convolutional neural network layers

2.2.1. Quaternion convolutional layer

In the quaternion convolutional layer, convolution is performed by convolving a quaternion filter matrix with a quaternion input vector. Let $\mathbf{W} = \mathbf{W}_0 + \mathbf{W}_1\boldsymbol{i} + \mathbf{W}_2\boldsymbol{j} + \mathbf{W}_3\boldsymbol{k}$ be a quaternion filter matrix and $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1\boldsymbol{i} + \mathbf{x}_2\boldsymbol{j} + \mathbf{x}_3\boldsymbol{k}$ be a quaternion input vector. The quaternion convolution between $\mathbf{W}$ and $\mathbf{x}$ is given by

$$\begin{aligned} \mathbf{W} \otimes \mathbf{x} = &(\mathbf{W}_0 * \mathbf{x}_0 - \mathbf{W}_1 * \mathbf{x}_1 - \mathbf{W}_2 * \mathbf{x}_2 - \mathbf{W}_3 * \mathbf{x}_3) \\ &+(\mathbf{W}_0 * \mathbf{x}_1 + \mathbf{W}_1 * \mathbf{x}_0 + \mathbf{W}_2 * \mathbf{x}_3 - \mathbf{W}_3 * \mathbf{x}_2)\boldsymbol{i} \\ &+(\mathbf{W}_0 * \mathbf{x}_2 - \mathbf{W}_1 * \mathbf{x}_3 + \mathbf{W}_2 * \mathbf{x}_0 + \mathbf{W}_3 * \mathbf{x}_1)\boldsymbol{j} \\ &+(\mathbf{W}_0 * \mathbf{x}_3 + \mathbf{W}_1 * \mathbf{x}_2 - \mathbf{W}_2 * \mathbf{x}_1 + \mathbf{W}_3 * \mathbf{x}_0)\boldsymbol{k} \end{aligned}, \tag{6}$$

where $*$ is real-valued convolution.

2.2.2. Quaternion batch normalization layer

Batch normalization layer is usually used to speed up training in a real-valued CNN [40]. In some cases, batch normalization is essential to train a model. So, quaternion batch normalization (QBN) proposed in [39] is also utilized for QCNN. The QBN composes of two steps presented in the following.

Firstly, a whitening approach is used to normalize the input data $\mathbf{x}$. The whitening approach is

performed through multiplying 0-centered data ($\mathbf{x} - \mathrm{E}[\mathbf{x}]$) by $\mathbf{W}$

$$\hat{\mathbf{x}} = \mathbf{W}(\mathbf{x} - \mathrm{E}[\mathbf{x}]), \tag{7}$$

where $\mathrm{E}[\mathbf{x}]$ is the mean of $\mathbf{x}$ and $\mathbf{W}$ is the covariance matrix of $\mathbf{x}$.

Secondly, two learnable parameters are introduced to make sure that the transformation inserted in this layer can represent the identity transform. The two learnable parameters β and γ scale and shift the normalized value as follow

$$\mathrm{BN}(\mathbf{x}) = \gamma\hat{\mathbf{x}} + \beta . \tag{8}$$

2.2.3. Other layers

Other layers in QCNN models, such as quaternion activation layers, quaternion pooling layers and quaternion dropout layer, are obtained by a so-called split approach [25] from the corresponding real-valued layers. Taking quaternion ReLU activation function for example, the quaternion ReLU activation function ζ is obtained by applying separate real-valued ReLU on all four parts of a quaternion vector $\mathbf{x} = \mathbf{x}_0 + \mathbf{x}_1\boldsymbol{i} + \mathbf{x}_2\boldsymbol{j} + \mathbf{x}_3\boldsymbol{k}$ as follow

$$\zeta(\mathbf{x}) = \varphi(\mathbf{x}_0) + \varphi(\mathbf{x}_1)\boldsymbol{i} + \varphi(\mathbf{x}_2)\boldsymbol{j} + \varphi(\mathbf{x}_3)\boldsymbol{k} , \tag{9}$$

where φ is the real-valued ReLU function.

3. Proposed algorithm

In this section, firstly, QFCN is proposed to generalize the real-valued FCN to quaternion field. Then, the SP-enhanced pairwise CRF used in the proposed color image splicing localization algorithm is described. Finally, the main architecture of the proposed algorithm is presented.

3.1. Quaternion fully convolutional neural networks(QFCNs)

FCN [41] is a special type of CNN with only convolutional layers. There are three commonly-used versions FCNs (FCN32, FCN16, FCN8) cast from VGG 16 [33] with different up-sampling layers. Taking FCN32 for example, the input image is processed by seven convolutional blocks to generate feature maps. Then, a 1×1 convolutional kernel is considered to predict scores for each class. Finally, a deconvolutional layer is used to up-sample coarse outputs to pixel-dense predictions.

It is obvious that the generated feature maps after seven convolutional blocks can affect the final results greatly. So, in this paper, the quaternion convolutional blocks given in subsection 2.2.1 are used to replace the real-valued convolutional blocks in FCN. In addition, the input image is represented by QR. The architectures of the original FCN32 and the proposed QFCN32 are shown in Figure 1. Following the construction of QFCN32, QFCN16 and QFCN8 are easy to build.

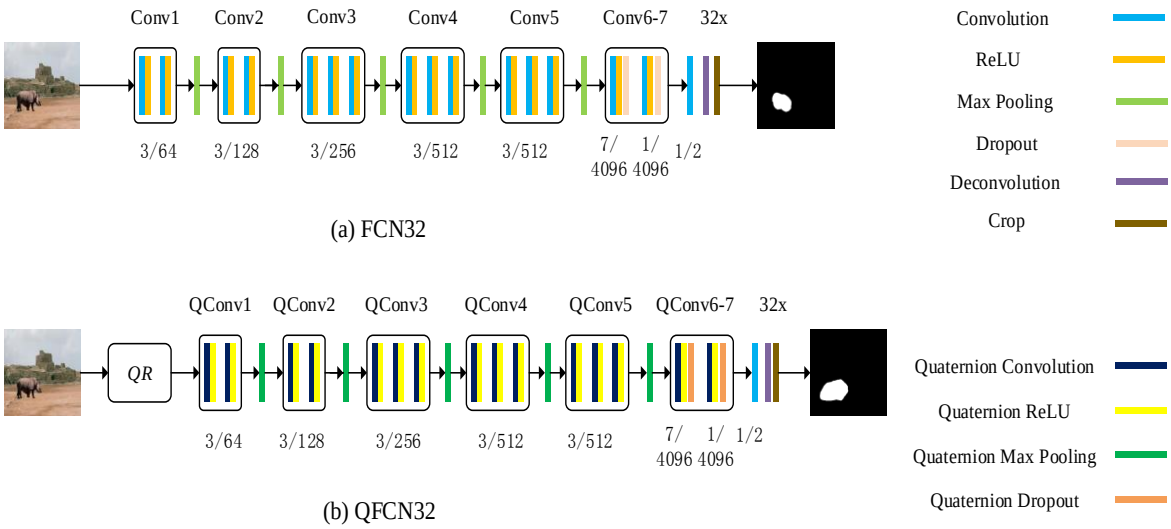

Figure 1. Architectures of FCN32 and QFCN32. Numbers in the form *m*/*n* refer to the kernel size *m* and the number of kernels *n* in the convolutional layer.

3.2. SP-enhanced pairwise CRF

In [22,23], the pairwise CRF is used to refine the results of FCN. CRF is a probabilistic graphical model which formulates label assignment problem as a probabilistic inference problem. It assigns similar pixels same label by capturing consistency between pixels. However, the pairwise CRF used in [22,23] only considers unary and pairwise potentials. It is not expressive enough to model higher level consistency, such as region-level consistency, co-occurrence of objects or detector-based cues [42,43].

In order to capture region-level consistency, Sulimowicz et al. [44] first introduced SP-enhanced pairwise potentials, and then proposed SP-enhanced pairwise CRF by combining conventional potentials used in pairwise CRF and their SP-enhanced pairwise potentials. The SP-enhanced pairwise potentials incorporate superpixel-based higher-order cues by conditioning on a superpixel segmentation image, which is obtained by an unsupervised segmentation algorithm, i.e., mean-shift algorithm. The mean-shift algorithm works by clustering pixels on the basis of low level image features [45]. For different images, the numbers of superpixels obtained by this algorithm are usually different. Furthermore, Sulimowicz et al. [44] proved theoretically that the sum of SP-enhanced pairwise potentials inside each superpixel was equal to robust superpixel-based CRF model proposed in [45]. Therefore, the SP-enhanced pairwise CRF is also a robust superpixel-based model. Experimental results presented in [44] show that the SP-enhanced pairwise CRF achieves a better performance than the pairwise CRF. So, in this paper, the SP-enhanced pairwise CRF is also used to refine the results of QFCN.

3.3. Main architecture of the proposed algorithm

The main architecture of the proposed algorithm is shown in Figure 2.

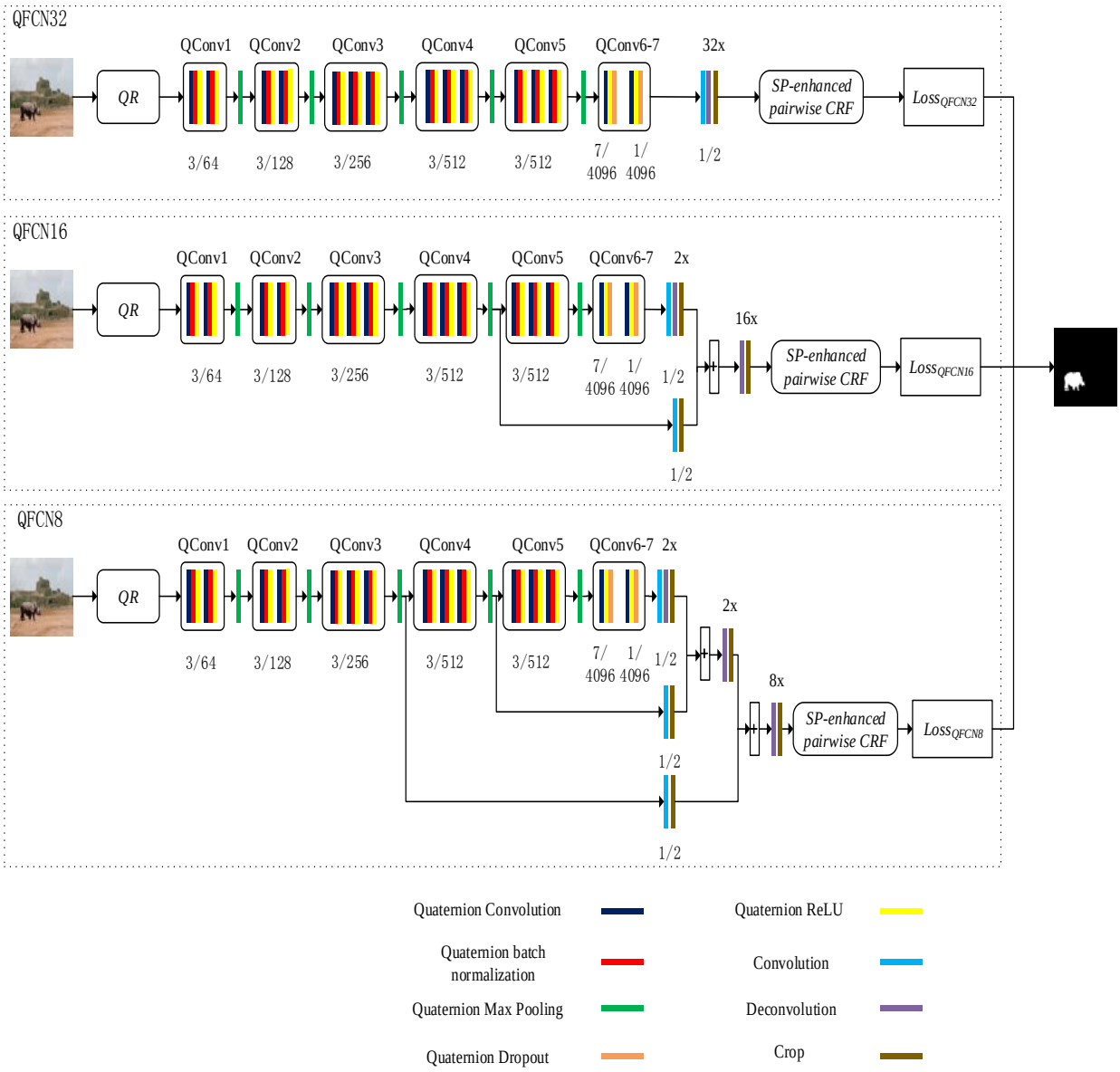

Figure 2. Main architecture of the proposed algorithm.

The proposed algorithm uses QFCN to predict splicing location map and considers three QFCNs with different up-sampling layers: QFCN32, QFCN16 and QFCN8. The reason of considering three networks is that one network can be specialized in handling one aspect of the whole problem, while the fusion of three networks can deal with different scales of image contents [22]. The SP-enhanced pairwise CRF is utilized for all three networks to improve the results obtained from QFCNs. In addition, quaternion batch normalization is also used to make the network converge easily. Finally, the final location map is obtained by merging the predictions of three networks. For each pixel, let y_{32}, y_{16}, and y_8 denote the predictions of three networks with the value 0 or 1. 0 is for unforged, and 1 is for forged. The final predictions m of this pixel is given by

$$m = \begin{cases} 1 & y_8 + y_{16} + y_{32} \geq 2 \\ 0 & else \end{cases}, \tag{10}$$

Eq. (10) shows that a pixel is forged if there are at least two networks predict this pixel as forged.

The details of the proposed algorithm are as follows：

(a) Taking QFCN32 for example, an input image is first represented by QR in Eq. (5). Then, seven quaternion convolutional blocks process the quaternionic input to generate feature maps. Both of the first two blocks consist of two 3×3 quaternion convolutional layers, each of which is followed by a quaternion ReLU layer and a quaternion batch normalization layer. The following three blocks are similar to the first two but having three 3×3 quaternion convolutional layers. Moreover, each of the first five blocks is followed by a quaternion pooling layer. The last two blocks have a quaternion convolutional layer followed by a quaternion ReLU layer and a quaternion dropout layer. The kernel sizes of the last two blocks are 7×7 and 1×1, respectively. In addition, the numbers of kernels in seven blocks are 64, 128, 256, 512, 512, 4096, and 4096, respectively.
(b) The generated maps are fed into a 1×1 convolutional layer to predict score for each class. Then a deconvolutional layer is used to up-sample the coarse outputs to pixel-dense predictions. Finally, a SP-enhanced pairwise CRF layer is used to refine the result.
(c) QFCN8 and QFCN16 are similar to QFCN32. The difference is that they have different up-sampling layers. QFCN16 fuses the results of the fifth and fourth quaternion pooling layers before deconvolutional layer, while QFCN8 combines the results the fifth, fourth and third quaternion pooling layers.
(d) The final predictions are obtained from three predictions from QFCN8, QFCN16 and QFCN32 by Eq. (10).

4. Experimental results and analysis

In this section, we compare the proposed color image splicing localization algorithm with some state-of-the-art algorithms on three publicly available datasets. All the deep learning-based algorithms are performed in Keras with 11GB GeForce GTX 1080 Ti, 3.20 GHz i7-6900K CPU, and 65GB RAM. The conventional algorithms are in Matlab.

4.1. Experimental datasets

In this paper, in order to evaluate the performance of the proposed algorithm, three datasets are considered: CASIA v1.0, CASIA v2.0 [46], and Columbia color DVMM [47]. CASIA v1.0 is a forgery dataset which focuses on color image splicing. The tampered regions are carefully selected and some post-processing operations are also applied. This dataset is composed of 1721 images with 384×256 or 256×384 resolution. The number of authentic and forged images is 183 and 180, respectively. CASIA v2.0 is an extended version of CASIA v1.0 with more forged images and more post-processing operations. It contains 7491 authentic images and 5123 spliced images with the resolution from 240×160 to 900×600. DVMM is the first publicly available color image dataset for image forgery detection and localization without editing or post-processing. This dataset contains 183 authentic images and 180 spliced images with the resolution from 757×568 to 1152×768. Notice that CASIA v1.0 and CASIA v2.0 do not provide ground truth masks. So, we use Adobe Photoshop software to generate the ground truth masks from the corresponding host images. Some

forged images and their corresponding ground truth masks are illustrated in Figure 3.

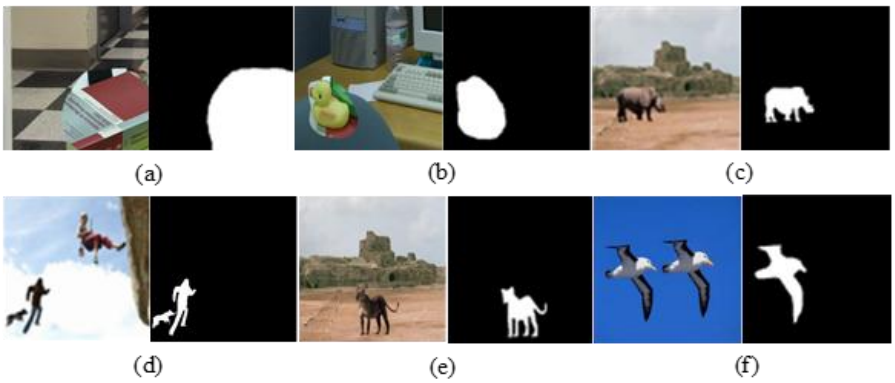

Figure 3. Some examples of three datasets and their corresponding ground truth masks. (a) and (b) are from DVMM dataset, (c) and (d) from CASIA v1.0 dataset, (e) and (f) from CASIA v2.0 dataset.

4.2. Evaluation metric

The accuracy of splicing localization is evaluated by the following per-pixel metric F-measure

$$F = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \tag{11}$$

Here *Precision* means the probability that a detected forgery is truly forged, while *Recall* represents the probability that a forgery is detected. They are defined by

$$Precision = \frac{T_P}{T_P + F_P}, \quad \text{and} \quad Recall = \frac{T_P}{T_P + F_N}. \tag{12}$$

where T_P is the number of correctly detected pixels, F_N means the number of missed forged pixels, and F_P represents the number of pixels erroneously detected as forged.

4.3. Experimental results and analysis

In order to evaluate the efficiency of the proposed QFCNs over the conventional real-valued FCNs, the first experiment directly trains three QFCNs (QFCN32, QFCN16 and QFCN8) and FCNs (FCN32, FCN16 and FCN8) respectively for splicing localization on CASIA v2.0 dataset. Notice that the SP-enhanced pairwise CRF is not considered in this experiment because we want to compare QFCNs and FCNs directly. In the experiment, we randomly select 5/6 spliced images to train the models and use the remaining images to evaluate the models. The average F-measure values of three QFCN-based algorithms and three FCN-based ones are given in Figure 4. It can be observed from Figure 4 that all the QFCN-based algorithms are superior to their corresponding FCN-based algorithms with same up-sampling layers. This is because the quaternion convolution can obtain more representative features than the conventional real-valued convolution [38].

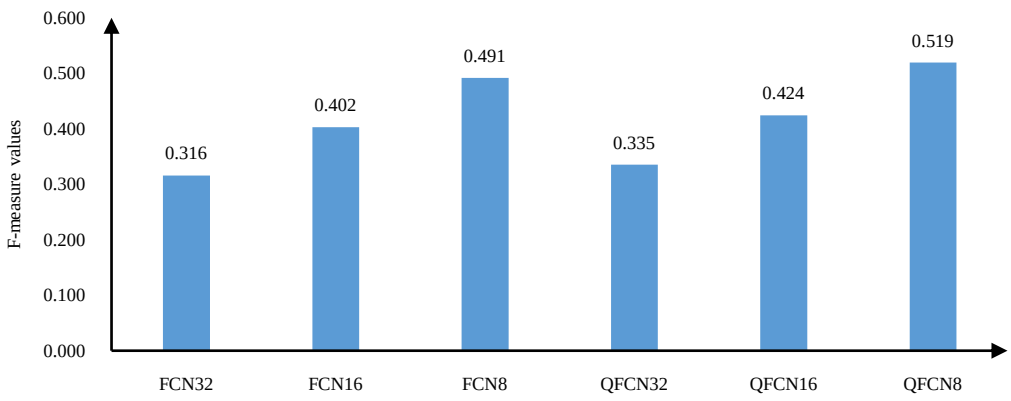

Figure 4. Comparison of QFCNs and FCNs in CASIA v2.0.

The second experiment is to test the influence of the parameters of mean-shift algorithm in the SP-enhanced pairwise CRF. In detail, kernel bandwidth parameter $h = (h_s, h_r)$ in the mean-shift algorithm has been considered. Here, h_s is the kernel bandwidth for spatial domain, and h_r is for range domain. The same experimental dataset used in the previous experiment is considered. The average F-measure values of the proposed algorithm with different parameters are given in Table 1. The results in Table 1 show that the kernel bandwidth parameter has little influence on the proposed algorithm and the optimal parameter is (8, 8) among ten compared parameters. So, the parameter (8, 8) is considered in the following experiments.

Table 1. Comparison of the proposed algorithm with different parameters in CASIA v2.0.

(h_s, h_r)	(8, 4)	(8, 8)	(8, 16)	(16, 4)	(16, 8)	(16, 16)	(32, 4)	(32, 8)	(32, 16)	(32, 32)
F-measure value	0.678	0.690	0.680	0.689	0.683	0.677	0.687	0.682	0.682	0.680

The third experiment is to evaluate the localization ability of the proposed algorithm based on QFCNs and SP-enhanced pairwise CRF. The same experimental dataset considered in the first experiment is used for this experiment. We compare the proposed algorithm with some existing algorithms, including eight conventional algorithms and five deep learning-based algorithms. Eight conventional algorithms contain NOI1 [4], NOI2 [5], CFA1 [6], CFA2 [7], ADQ [9], NADQ [10], DCT [11] and BLK [12]. They are implemented through a publicly available Matlab toolbox written by Zampoglou et al. and Xiao et al. [48,49]. Five deep learning-based algorithms are FCN+CRF [22], MFCN [21], QFCN+CRF, FCN+ResNet+CRF and LSTM+EnDec [24], respectively. QFCN+CRF uses QFCN to replace FCN in the work FCN+CRF [22]. The main objective of the comparison between FCN+CRF and QFCN+CRF is to show the improvement by using QFCN. FCN+ResNet+CRF combines ResNet [37] with the work FCN+CRF [22]. The comparison of these fourteen algorithms by average F-measure values are given in Figure 5. It can be seen from Figure 5 that: (a) all the deep learning-based algorithms outperform all the conventional algorithms. It is owing to the fact that the deep learning-based algorithms learn the feature automatically and expectedly; (b) among six deep learning-based algorithms, the proposed algorithm achieves the best performance. It is better than QFCN+CRF because the SP-enhanced pairwise CRF used in the proposed algorithm is more effective than the CRF in enforcing long-range consistency in pixel-wise

labeling problems [44]. The proposed algorithm performs better than FCN+CRF [22] due to the use of quaternion-based method and SP-enhanced pairwise CRF. MFCN [21] is superior to FCN+CRF [22] and QFCN+CRF because the boundaries of spliced regions are also trained.

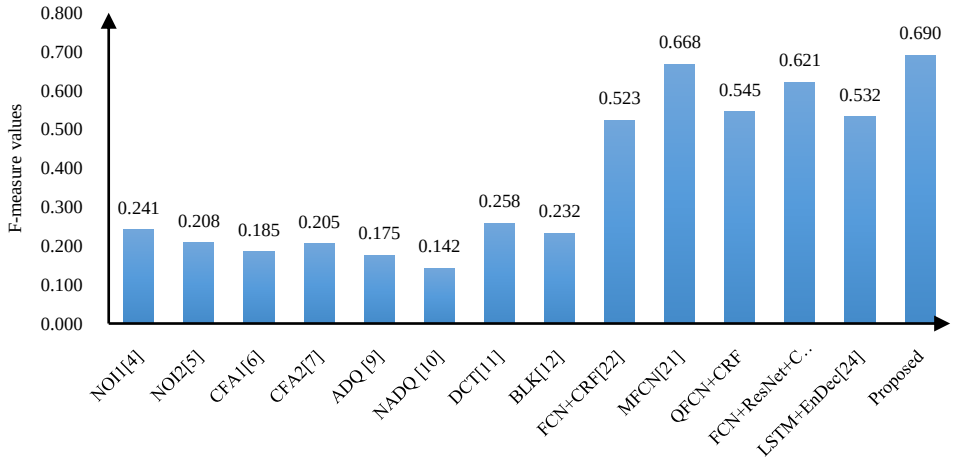

Figure 5. Comparison of localization ability in CASIA v2.0.

The fourth experiment is to evaluate the generalization ability of the proposed algorithm. In this experiment, all spliced images in CASI v2.0 dataset are used for training, while CASIA v1.0 dataset and DVMM dataset are for testing. Figure 6 shows the average F-measure values of all the algorithms. It can be observed from Figure 6 that the proposed algorithm and FCN+ResNet+CRF achieve the best performance in both datasets. In addition, some conventional algorithms also have a good performance in DVMM dataset. The main reason is that the DVMM dataset does not perform post-processing after being tampered and does not contain small spliced regions.

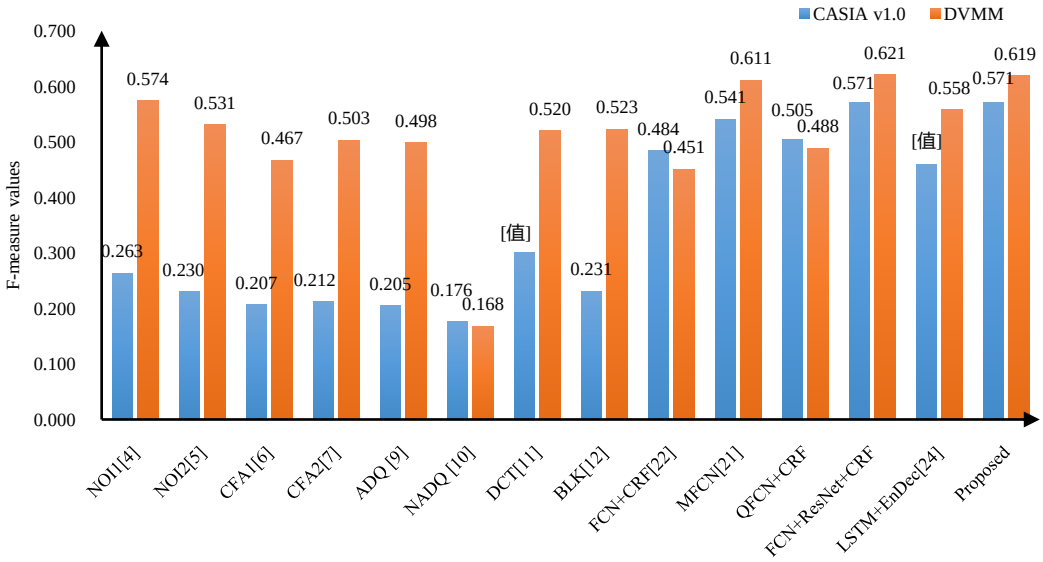

Figure 6. Comparison of generalization ability in CASIA v1.0 and DVMM.

The last experiment is to evaluate the robustness against JPEG compression, Gaussian blur and Gaussian noise. Same as the previous experiment, all spliced images in CASIA v2.0 dataset are utilized for training, while all spliced images in CASIA v1.0 dataset are used for testing. For the JPEG compression, the quality factors are set to two levels: QF = 50 and QF = 70. For Gaussian blur operation, the Gaussian smoothing kernel (σ) varies from 0.5 to 2.0 with a step size of 0.5. For Gaussian noise addition, the SNR value varies from 25 db to 15 db with a step size of −5. The comparison results by average F-measure values for three attacks are given in Table 2. The results in Table 2 show that, similar to the previous two experiments, the proposed algorithm performs best among fourteen compared algorithms in all the three types of attacks with different levels. In addition, the performance of all algorithms decreases with the increase of attack intensity.

Table 2. Comparison of robustness against three types of attacks in CASIA v1.0.

Algorithms	JPEG compression		Gaussian blur				Gaussian noise		
	QF=70	QF=50	σ =0.5	σ =1.0	σ =1.5	σ =2.0	SNR=25dB	SNR=20dB	SNR=15dB
NOI1[4]	0.237	0.241	0.260	0.253	0.248	0.248	0.233	0.229	0.216
NOI2[5]	0.212	0.206	0.228	0.226	0.223	0.219	0.228	0.219	0.221
CFA1[6]	0.200	0.198	0.205	0.200	0.200	0.199	0.206	0.198	0.204
CFA2[7]	0.196	0.204	0.210	0.208	0.208	0.207	0.182	0.185	0.184
ADQ [9]	0.199	0.196	0.204	0.198	0.192	0.157	0.205	0.199	0.201
NADQ [10]	0.180	0.170	0.175	0.153	0.154	0.150	0.170	0.168	0.150
DCT[11]	0.312	0.326	0.300	0.296	0.298	0.286	0.288	0.298	0.287
BLK[12]	0.216	0.220	0.231	0.229	0.229	0.229	0.226	0.227	0.213
FCN+CRF[22]	0.484	0.481	0.484	0.482	0.478	0.475	0.482	0.482	0.471
MFCN[21]	0.541	0.532	0.538	0.537	0.535	0.524	0.541	0.530	0.524
QFCN+CRF	0.504	0.502	0.504	0.502	0.498	0.494	0.502	0.500	0.493
FCN+ResNet+CRF	0.557	0.546	0.551	0.550	0.543	0.542	0.560	0.556	0.546
LSTM+EnDec[24]	0.430	0.408	0.450	0.436	0.425	0.401	0.409	0.401	0.394
Proposed	0.568	0.566	0.569	0.565	0.560	0.555	0.562	0.559	0.553

In order to better show the superior performance of the proposed algorithm, Figure 7 presents the visual results and their corresponding F-measure values of the proposed algorithm and the other five deep learning-based algorithms. These results are corresponding to the forged images given in Figure 3. It can be observed from Figure 7 that: (a) the visual comparison is basically in keeping with the F-measure value comparison presented in Figure 5, Figure 6 and Table 2; (b) the proposed algorithm locates the forged regions more accurate, especially for CASIA v1.0 and CASIA v2.0 datasets. For example, the proposed algorithm can detect the legs of animals accurately for the forged images Figure 3 (c) in CASIA v1.0 and Figure 3 (e) in CASIA v2.0. While it is not the case for other algorithms, which can only get the rough results; (c) the difference among six compared algorithms on DVMM dataset is relatively small. Because the DVMM dataset does not perform post-processing and the splicing areas are usually not small ones. So, the proposed algorithm outperforms other algorithms both in F-measure values and visual results.

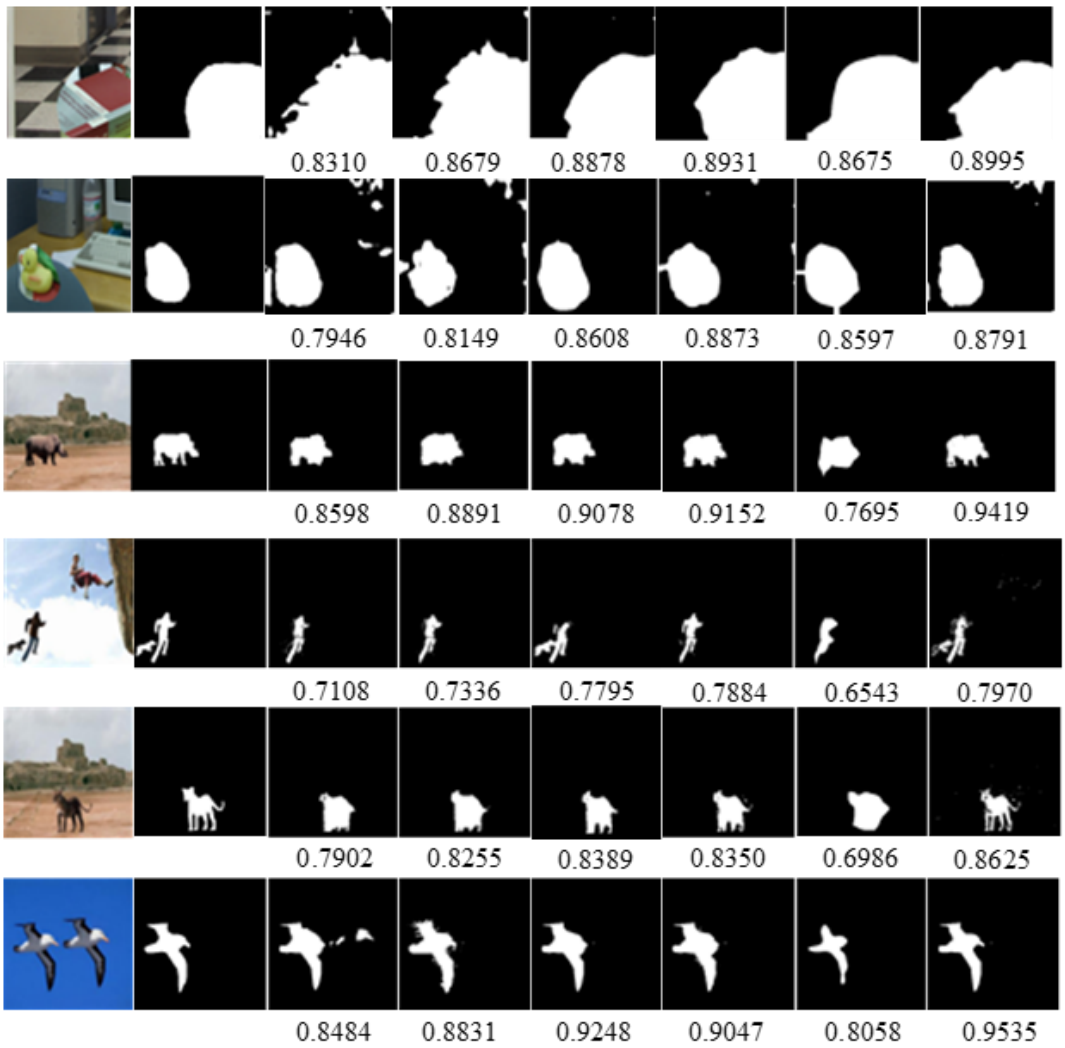

Figure 7. Visual comparison of four deep learning-based algorithms. For rows, from top to down, the results for Figure 3 (a)-(f). For columns, from left to right, spliced image, ground truth mask, localization results for FCN+CRF [22], QFCN+CRF, MFCN [21], FCN+ResNet+CRF, LSTM+EnDec [24] and the proposed algorithm.

5. Conclusions

In this paper, QFCNs are proposed to extend real-valued FCNs to quaternion field. In addition, a color image splicing localization algorithm based on QFCNs and SP-enhanced pairwise CRF is proposed. The proposed algorithm is superior to some existing algorithms for the following reasons: (a) compared with the conventional algorithms without deep learning, the proposed algorithm is a deep learning-based algorithm. It integrates feature extraction and localization map generation into the network for end-to-end training. In addition, it learns effective features automatically during the training; (b) compared with other deep learning-based algorithms, the proposed algorithm uses the quaternion-based color image processing method to capture the integrity of three channels and the inherent correlation between color channels; (c) SP-enhanced pairwise CRF is used to refine the results obtained by QFCNs. For the future work, we will try to construct a network to model sensor noise well and then use it for image splicing localization.

Acknowledgments

This work was supported in part by NSFC under Grants 61572258, 61771231, 61772281, and 61672294, in part by the PAPD fund, and in part by Qing Lan Project.

Conflict of interest

The authors declare no conflict of interest.

References

1. G. K. Birajdar and V. H. Mankar, Digital image forgery detection using passive techniques: A survey, *Digit. Invest.*, **10**(2013), 226–245.
2. B. J. Chen, M. Yu, Q. T. Su, et al., Fractional quaternion cosine transform and its application in color image copy-move forgery detection, *Multimed. Tools Appl.*, (2018), 1–17.
3. C. M. Pun, B. Liu and X. C. Yuan, Multi-scale noise estimation for image splicing forgery detection, *J. Vis. Commun. Image Represent.*, **10**(2016), 195–206.
4. B. Mahdian and S. Saic, Using noise inconsistencies for blind image forensics, *Image Vis. Comput.*, **27**(2009), 1497–1503.
5. S. Lyu, X. Pan and X. Zhang, Exposing region splicing forgeries with blind local noise estimation, *Int. J. Comput. Vis.*, **110**(2014), 202–221.
6. P. Ferrara, T. Bianchi, R. A. De, et al., Image forgery localization via fine-grained analysis of CFA artifacts, *IEEE Trans. Inf. Forensic Secur.*, **10**(2013), 226–245.
7. A. E. Dirik and N. Memon, Image tamper detection based on demosaicing artifacts, *In: IEEE International Conference on Image Processing*, (2009), 1497–1500.
8. E. González, A. Sandoval, L. García, et al., Digital image tamper detection technique based on spectrum analysis of CFA artifacts, *Sensors*, **18**(2018), 2804.
9. Z. Lin, J. He, X. Tang, et al., Fast, automatic and fine-grained tampered JPEG image detection via DCT coefficient analysis, *Pattern Recognit.*, **42**(2009), 2492–2501.
10. T. Bianchi and A. Piva, Image forgery localization via block-grained analysis of JPEG artifacts, *IEEE Trans. Inf. Forensic Secur.*, **7**(2012), 1003–1017.
11. S. M. Ye, Q. Sun and E. C. Chang, Detecting digital image forgeries by measuring inconsistencies of blocking artifact, *In: IEEE International Conference on Multimedia and Expo*, (2007), 12–15.
12. W. Li, Y. Yuan and N. Yu, Passive detection of doctored JPEG image via block artifact grid extraction, *Signal Process.*, **89**(2009), 1821–1829.
13. W. Luo, J. Huang and G. Qiu, JPEG error analysis and its applications to digital image forensics, *IEEE Trans. Inf. Forensic Secur.*, **5**(2010), 480–491.
14. F. Huang, J. Huang and Y. Q. Shi, Detecting double JPEG compression with the same quantization matrix, *IEEE Trans. Inf. Forensic Secur.*, **5**(2010), 848–856.
15. A. C. Popescu and H. Farid, Exposing digital forgeries by detecting traces of resampling, *IEEE Trans. Signal Process.*, **53**(2005), 758–767.
16. H. D. Li, W. Q. Luo, X. Q. Qiu, et al., Image forgery localization via integrating tampering possibility maps, *IEEE Trans. Inf. Forensic Secur.*, **12**(2017), 1240–1252.

17. D. Cozzolino, G. Poggi and L. Verdoliva, Recasting residual-based local descriptors as convolutional neural networks: an application to image forgery detection, *In: ACM Workshop on Information Hiding and Multimedia Security*, (2017), 159–164.
18. Y. Liu, Q. Guan, X. Zhao, et al., Image forgery localization based on multi-scale convolutional neural networks, *In: ACM Workshop on Information Hiding and Multimedia Security*, (2018), 85–90.
19. J. H. Bappy, A. K. Roy, J. Bunk, et al., Exploiting spatial structure for localizing manipulated image regions, *In: IEEE International Conference on Computer Vision*, (2017), 4970–4979.
20. Z. Shi, X. Shen and H. Kang, Image manipulation detection and localization based on the dual-domain convolutional veural networks, *IEEE Access*, **6**(2018), 76437–76453.
21. R. Salloum, Y. Ren and C. C. J. Kuo, Image splicing localization using a multi-task gully convolutional network (MFCN), *J. Vis. Commun. Image Represent.*, **51**(2018), 201–209.
22. B. Liu and C. M. Pun, Locating splicing forgery by fully convolutional networks and conditional random field, *Signal Process.Image Commun.*, **66**(2018), 103–112.
23. B. J. Chen, X. M. Qi, Y. T. Wang, et al., An Improved Splicing Localization Method by Fully Convolutional Networks, *IEEE Access*, **6**(2018), 69472–69480.
24. J. H. Bappy, C. Simons, L. Nataraj, et al., Hybrid LSTM and encoder-decoder architecture for detection of image forgeries, *IEEE Trans. Image Process.*, **28**(2019), 3286–3300.
25. T. Parcollet, M. Morchid and G. Linarès, Quaternion convolutional neural networks heterogeneous image processing, preprint, arXiv: 1811.02656.
26. T. A. Ell and S. J. Sangwine, Hypercomplex fourier transforms of color images, *IEEE Trans. Image Process.*, **16**(2007), 22–35.
27. B. J. Chen, G. Coatrieux, G. Chen, et al., Full 4-D quaternion discrete Fourier transform based watermarking for color images, *Digit. Signal Proc.*, **28**(2014), 106–119.
28. N. Matsui, T. Isokawa, H. Kusamichi, et al., Quaternion neural network with geometrical operators, *J. Intell. Fuzzy Syst.*, **15**(2004), 149–164.
29. X. Xu, Z. Guo, C. Song, et al., Multispectral palmprint recognition using a quaternion matrix, *Sensors*, **12**(2012), 4633–4647.
30. B. J. Chen, J. H. Yang, B. Jeon, et al., Kernel quaternion principal component analysis and its application in RGB-D object recognition, *Neurocomputing*, **266**(2017), 293–303.
31. G. L. Xu, X. T. Wang and X. G. Xu, Fractional quaternion Fourier transform, convolution and correlation, *Signal Process.*, **88**(2008), 2511–2517.
32. B. J. Chen, C. F. Zhou, B. Jeon, et al., Quaternion discrete fractional random transform for color image adaptive watermarking, *Multimed. Tools Appl.*, **77**(2018), 20809–20837.
33. K. Simonyan and A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv: 1409.1556.
34. Q. Cui, S. McIntosh and H. Y. Sun, Identifying materials of photographic images and photorealistic computer generated graphics based on deep CNNs, *Comput. Mat. Contin.*, **55**(2018), 229–241.
35. H. Y. Zhao, C. Che, B. Jin, et al., A viral protein identifying framework based on temporal convolutional network, *Math. Biosci. Eng.*, **16**(2019), 1709–1717.
36. L. G. Zheng and C. Song, Fast near-duplicate image detection in Riemannian space by a novel hashing scheme, *Comput. Mat. Contin.*, **56**(2018), 529–539.

37. K. He, X. Zhang, S. Ren, et al., Deep residual learning for image recognition, *In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, (2016), 770–778.
38. X. Zhu, Y. Xu and H. Xu, Quaternion convolutional neural networks, *In: European Conference on Computer Vision*, (2018), 631–647.
39. C. J. Gaudet and A. S. Maida, Deep quaternion networks, *In: IEEE International Joint Conference on Neural Networks*, (2018), 1–8.
40. S. Ioffe and C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift, *In: International Conference on International Conference on Machine Learning*, (2015), 448–456.
41. E. Shelhamer, J. Long and T. Darrell, Fully Convolutional Networks for Semantic Segmentation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39**(2014), 640–651.
42. A. Arnab, S. Jayasumana, S. Zheng, et al., Higher order conditional random fields in deep neural networks, *In: European Conference on Computer Vision*, (2016), 524–540.
43. Y. H. Zheng, L. Sun, S. F. Wang, et al., Spatially regularized structural support vector machine for robust visual tracking. *IEEE Trans. Neural Netw. Learn. Syst.*, 2018. DOI: 10.1109/tnnls.2018.2855686
44. L. Sulimowicz, I. Ahmad and A. Aved, Superpixel-enhanced pairwise conditional random field for semantic segmentation, *In: IEEE International Conference on Image Processing*, (2018), 271–275.
45. P. Kohli and P. H. S. Torr, Robust higher order potentials for enforcing label consistency, *Int. J. Comput. Vis.*, **82**(2009), 302–324.
46. J. Dong and W. Wang, CASIA tampered image detection evaluation (TIDE) database, v1.0 and v2.0, Chinese Academy of Sciences, 2011, Available online: http://forensics.idealtest.org/.
47. T. T. Ng and S. F. Chang, A dataset of authentic and spliced image blocks, Columbia University, 2004, Available online: http://www.ee.columbia.edu/ln/dvmm/downloads/.
48. M. Zampoglou, S. Papadopoulos and Y. Kompatsiaris, Large-scale evaluation of splicing localization algorithms for web images, *Multimed. Tools Appl.*, **76**(2017), 1–34.
49. F. Xiao, L.Chen, H. Zhu, et al., Anomaly-tolerant network traffic map estimation via noise-immune temporal matrix completion, *IEEE J. Sel. Area. Comm.*, **37**(2019), 1192–1204.