Quaternion Generative Adversarial Networks for Inscription Detection in Byzantine Monuments

Giorgos Sfikas[1], Angelos P. Giotis[1], George Retsinas[2], and Christophoros Nikou[1]

[1] Department of Computer Science and Engineering
University of Ioannina, Greece
{sfikas,agiotis,cnikou}@cse.uoi.gr
[2] School of Electrical and Computer Engineering
National Technical University of Athens, Greece
{gretsinas@central.ntua.gr}@ece.ntua.gr

Abstract. In this work, we introduce and discuss Quaternion Generative Adversarial Networks, a variant of generative adversarial networks that uses quaternion-valued inputs, weights and intermediate network representations. Quaternionic representation has the advantage of treating cross-channel information carried by multichannel signals (e.g. color images) holistically, while quaternionic convolution has been shown to be less resource-demanding. Standard convolutional and deconvolutional layers are replaced by their quaternionic variants, in both generator and discriminator nets, while activations and loss functions are adapted accordingly. We have succesfully tested the model on the task of detecting byzantine inscriptions in the wild, where the proposed model is on par with a vanilla conditional generative adversarial network, but is significantly less expensive in terms of model size (requires $4\times$ less parameters).

Keywords: Quaternions · Generative Adversarial Networks · Byzantine inscriptions · Text detection

1 Introduction

Digitization and online accessibility in cultural institutions such as museums, libraries and archives can achieve much greater visibility to the public when the digitized content is organized in meaningful entities. For example, text in natural images generally conveys rich semantic information about the scene and the enclosed objects, which might be of great use in real scenarios where the digitized raw image information is not directly exploitable for searching and browsing.

One of the most prominent trends in content-based image retrieval applications is to discriminate which part of the image includes useful information, as opposed to background objects, occlusion and task-irrelevant parts [20]. Such tasks may concern image analysis, understanding, indexing or classification of objects according to some inherent property. In the particular case of text understanding applications, the main goal is to retrieve regions that contain solely

textual cues, either as holistic region information or as textual parts at line, word or even character level.

Text detection is a challenging task due to the variety of text appearance, the unconstrained locations of text within the natural image, degradations of text components over hundreds of years, as well as the complexity of each scene. To address these challenges, standard convolutional neural networks (CNNs) have been the main attraction over the last five years for text detection [8, 23]. However, the effectiveness of CNNs is usually limited by the homogeneity of the dataset images used for training as well as the particular loss function that is to be minimized for the specific task at hand. Generative Adversarial Networks (GANs) [5] offer a more flexible framework that can in effect learn the appropriate loss function to satisfy the task at hand. GANs setup an adversarial learning paradigm where the game dynamics of two players-networks lead to a model that, in its convolutional variant is the state of the art in numerous vision tasks today.

Fig. 1. Sample images of our inscription dataset.

With the current work, we discuss a novel neural network variant that brings together the concepts of GANs with that of Quaternionic convolution and deconvolution, and build a model that can effectively perform text detection in a context where the content of interest is "donor" inscriptions found in byzantine monuments [9, 21] (see Fig. 1,2). Quaternions are a form of non-real numbers that can be understood as 4-dimensional generalizations of complex num-

bers, with one real part and three independent imaginary parts. The use of non-real numbers as neuron and parameter values has been proposed as far as 1991, with an adaptation of backpropagation for complex numbers [10]. Similar developments for quaternions have followed suit [14]. The more recently proposed quaternion convolutional neural networks (QCNNs) [18, 26] a special form of convolution that makes use of quaternion product rules, effectively treating multichannel information holistically. Furthermore, QCNNs have been shown to be much more economical (i.e. less resource-demanding) networks than their non-quaternionic counterparts, with four times smaller parameter set size [17, 18]. Motivated by the promising properties of quaternionic neural networks, we propose using quaternionic operations with adversarial networks. In particular, the contribution of this work concerns the introduction of quaternionic convolution to the conditional convolutional generative adversarial paradigm, where we replace the encoder-decoder architecture with quaternionic layer versions, and otherwise adapt network architecture where necessary. In order to setup our numerical and qualitative experiments, we test the proposed model for inscription localization in the wild. In terms of numerical results, we conclude that the proposed model attains comparable evaluation scores to its non-quaternionic counterpart, while being less resource demanding.

The remainder of this paper is structured in the following manner. In section 2, we review related work. In section 3, we present the basics of quaternion algebra, and in section 4, we move to quaternionic convolution and its use with convolutional neural networks. In section 5, we discuss the proposed model and we present the dataset, task and numerical experiments employed in section 6. Finally, with section 7, we close the paper and discuss future work.

2 Related work

The automatic detection of text can be categorized into two main families. The first direction includes identifying text of scanned document images whereas the second contains text captured by natural images (indoor or outdoor images with text of more complex shapes, cuneiform tablet images or inscriptions) which is further subject to various geometric distortions, illumination and environmental conditions. The latter category is also known as text detection in the wild or scene text detection [19]. In the first category, text detection in printed documents is usually tackled by OCR techniques [25], while in handwritten document images, the problem is formulated as a keyword search in a segmentation free scenario [4].

In the text detection-in-the-wild paradigm, conditions such as wide variety of colors and fonts, orientations and languages are present. Moreover, scene elements might have similar appearance to text components, and finally, images may be distorted with blurriness, or contain degradations due to low camera resolution during digitization process, capturing angle and partial occlusions. Under such adverse situations deep learning based methods have shown great effectiveness in detecting text. Recent deep approaches for text detection in the

Fig. 2. Example ground-truth annotation for selected samples from our inscription dataset.

wild, inspired by object detection frameworks, can be categorized into *bounding-box regression based*, *segmentation-based* and *hybrid* approaches.

Bounding-box regression based methods for text detection [11] regard text as an object and aim to predict the candidate bounding boxes directly. Segmentation-based methods in [24] enforce text detection as a semantic segmentation task, aiming to classify text regions at pixel level and then obtain bounding boxes containing text during post-processing. Hybrid methods [12] rely on a segmentation step to predict score maps of text which in turn yield text bounding-boxes as a result of regression. Similarly to [24], our method localizes text in a holistic manner, by performing text detection as a semantic segmentation problem to produce global pixel-wise prediction maps.

While CNNs are at the top of the dominant problem-solvers in image recognition tasks, such as the text detection in the wild case explored in this work, traditional real-valued CNNs encode local relations of the input features from R,G,B channels of each pixel along with structural relations composed by groups of pixels, independently. On the contrary, our proposed quaternionic conditional adversarial network treats text detection as a semantic segmentation task, performing at input RGB channels holistically with the use of quaternions, so as to obtain a binary output of white text pixels. To our knowledge, GANs have not been used yet for text detection in the wild [19]. Moreover, the quaterionic representation of the conditional variant of the generative adversarial networks is a first attempt to discriminate a text region by its non-text counterpart with less computational load.

Recent works on quaternion CNNs [18, 26] indicate that the lower number of parameters required for the multidimensional representation of a single pixel in R,G,B channels leads to better image classification results than traditional CNNs. The authors claim that the performance boost is also due to the specific quaternion algebra. Such a boost is further explored in [16], where instead of a real-valued dot product, a vector product operation allows quaternion CNNs to capture internal latent relations by sharing quaternion weights during the product operation, and in turn by creating relations within the product's elements.

3 Elements of Quaternions

Quaternions, introduced in the mid-19th century, form an algebraic structure known as a skew-field, that is characterized by all the properties of a field except that of multiplication commutativity. We denote the quaternion skew-field as $\mathbb{H}$. Quaternions are four-dimensional, in the sense of $\mathbb{H}$ being isomorphic to $\mathbb{R}^4$, and each $q \in \mathbb{H}$ can be written as:

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

where $a, b, c, d \in \mathbb{R}$ and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are independent imaginary units. Hence, analogous to the representation of complex numbers, which bear one real and one imaginary part, quaternions have one real and three independent imaginary parts. Alternatively, quaternions can be represented as the sum of a scalar (their real

part) and a three-dimensional vector (their imaginary part). Formally we can write:

$$q = S(q) + V(q), \tag{2}$$

where $S(q) = a$ and $V(q) = b\boldsymbol{i}+c\boldsymbol{j}+d\boldsymbol{k}$. Further generalizing the related $\boldsymbol{i}^2 = -1$ formula for complex numbers, for quaternions we have:

$$\boldsymbol{i}^2 = \boldsymbol{j}^2 = \boldsymbol{k}^2 = \boldsymbol{ijk} = -1,$$

$$\boldsymbol{ij} = -\boldsymbol{ji} = \boldsymbol{k}, \boldsymbol{jk} = -\boldsymbol{kj} = \boldsymbol{i}, \boldsymbol{ki} = -\boldsymbol{ik} = \boldsymbol{j}. \tag{3}$$

Quaternion conjugacy is defined as:

$$\bar{q} = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}, \tag{4}$$

while quaternion magnitude is defined as:

$$|q| = \sqrt{q\bar{q}} = \sqrt{\bar{q}q} = \sqrt{a^2 + b^2 + c^2 + d^2}. \tag{5}$$

As a consequence of the properties of a skew-field and eq. (3), we have the following multiplication rule for quaternions:

$$\begin{aligned} pq =& (a_p a_q - b_p b_q - c_p c_q - d_p d_q) + && (6)\\ & (a_p b_q + b_p a_q + c_p d_q - d_p c_q)\boldsymbol{i} + && (7)\\ & (a_p c_q - b_p d_q + c_p a_q + d_p b_q)\boldsymbol{j} + && (8)\\ & (a_p d_q + b_p c_q - c_p b_q + d_p a_q)\boldsymbol{k}, && (9) \end{aligned}$$

where $p = a_p + b_p\boldsymbol{i} + c_p\boldsymbol{j} + d_p\boldsymbol{k}$ and $q = a_q + b_q\boldsymbol{i} + c_q\boldsymbol{j} + d_q\boldsymbol{k}$. Following the notation of eq. (2), we can write the above rule also as:

$$pq = S(p)S(q) - V(p) \cdot V(q) + S(p)V(q) + S(q)V(p) + V(p) \times V(q), \tag{10}$$

where $\cdot$ and $\times$ denote the dot and cross product respectively. Interestingly, note that when p, q are pure (i.e., they have zero respective real parts), the quaternion product boils down to a cross product. The above formulae are also referred to as a Hamilton product [17] in the literature.

4 Quaternionic Convolutional Neural Networks

Quaternionic Convolutional Neural Networks have been recently introduced as variants of the widely used Convolutional Neural Networks that have quaternionic model parameters, inputs, activations, pre-activations and outputs. This creates issues with a number of network components and concepts, including the definition of convolution, whether standard activation functions are usable and how back-propagation is handled. In theory, multiple proposals for a convolution operation could be considered [2]. Two quaternionic extensions of convolution have been succesfully employed in two recent works [18, 26]. In all cases, a quaternionic kernel $g \in \mathbb{H}^{K \times K}$ acts on an input feature map $f \in \mathbb{H}^{M \times N}$ to generate

the output map $g \in \mathbb{H}^{M+K-1 \times N+K-1}$. The two extensions differ in the choice of elementary operation used in each case.

In [26], a convolution extension that is based on the equation used to apply quaternionic rotation is employed (i.e. $w \rightarrow qw\bar{q}$, where q is a pure unit quaternion). In particular, they define quaternionic convolution $g = f * w$ as:

$$g_{kk'} = \sum_{l=1}^{K} \sum_{l'=1}^{K} s_{ll'}^{-1} w_{ll'} f_{(k+l)(k'+l')} \bar{w}_{ll'}, \tag{11}$$

where $f = [f_{ij}]$ denotes the input feature map, $w = [w_{ij}]$ is the convolution kernel, and $s_{ll'} = |w_{ll'}|$.

In [18], which is the convolution version that we test in this work, convolution is more simply defined as:

$$g_{kk'} = \sum_{l=1}^{K} \sum_{l'=1}^{K} w_{ll'} f_{(k+l)(k'+l')}, \tag{12}$$

where the definition is analogous to standard convolution, with the difference that elements are quaternionic and the kernel multiplies the signal from the left on each summation term. Strided convolution, deconvolution and padding are also defined analogously to real-valued convolution.

Concerning activation functions, the most straightforward option is to use standard activations that are used in real-valued networks (e.g. sigmoid, ReLU, etc.) and use them on each quaternion real and imaginary part separately, as if they were separate real channels. This type of activations are referred to in the literature as split-activation functions. In this work, we use split-activation versions of leaky Rectified linear unit (ReLU) and the sigmoid function.

5 Proposed model

The proposed model is made up of the well-known pair of the generator and discriminator networks that are used in standard GANs. The vanilla (non-conditional) GAN objective function [13] is, in its original form as follows:

$$L_{\text{GAN}} = E_x \log D(x) + E_z \log(1 - D(G(z))), \tag{13}$$

where $G(\cdot)$ and $D(\cdot)$ denote the generator and discriminator network respectively. x are samples of the training set, while z denotes random noise that is used as input to the generator. For the discriminator, the aim is to maximize this function, while for the generator the aim is to minimize it. These competing terms result in a two-player game, of which we require to obtain a parameter set that would correspond to a Nash equilibrium.

We employ a supervised variant that is referred to as a conditional GAN (cGAN) architecture, made popular with the pix2pix model [7]. Formally, the objective function is written as:

$$L_{\text{cGAN}} = E_x[logD(y)] + E_x[log(1 - D(G(x)))]] + \lambda E_{x,y}[\|y - G(x)\|_1] \tag{14}$$

where we can comment on a number of differences comparing with the standard GAN formula of eq. (13). In particular, no random noise variable z exists, and on the contrary the generator takes as input a sample x to produce a target y. In that sense, the cGAN is supervised; a cGAN learns a mapping from input x to target y. Also, a second L_1 regularizing term is employed, penalizing the difference of the produced $G(x)$ to the desired target y. A regularizing term λ controls trade-off of the two terms.

In this work, x is a quaternion-valued image, formally $x \in \mathbb{H}^{H \times W}$, where H and W are image height and width in pixels. In particular x is assumed to be a dataset image, and estimate $G(x)$ is a detection heatmap that ranges in $[0, 1]$. A pixel value of $G(x)$ that is close to 1 means a high probability that this pixel is part of a text inscription, and vice-versa. Ground truth target y is binary, with values in $\{0, 1\}$ (see Fig. 2). In order to form each quaternion-valued input x, we assign each of its three colour channels (Red, Green, Blue) to each of the quaternion imaginary axes. Hence, we assign $Red \rightarrow \boldsymbol{i}$, $Green \rightarrow \boldsymbol{j}$, $Blue \rightarrow \boldsymbol{k}$. The real part is left to be equal to zero, or in other words all values of x are pure quaternions.

The generator is constructed as a U-net-like model [22] with two symmetric groups of layers, arranged to an encoder and a decoder part. The encoder is composed of strided quaternionic convolutional layers that produce quaternionic feature maps of progressively lower resolution in comparison to the original input image size. The decoder mirrors the encoder layers, by using a quaternionic deconvolutional layer for each forward convolution layer of the encoder, and upsampling feature maps progressively to the original resolution. Furthermore, U-net-like skip connections connect corresponding encoder - decoder layers. We use 4 quaternionic convolutional layers for the encoder, and 4 quaternionic deconvolutional layers for the decoder. Dropout layers top layers 5 and 6. Convolutions are strided with stride=2, kernel sizes=4×4, and output number of channels equal to $16, 32, 64, 64$ for layers 1 to 4 respectively. Deconvolutional layers share the same characteristics, mirroring the encoder architecture, with added skip connections. All layers, except the final layer, are topped by split-activation leaky ReLU functions with parameter = 0.2. These act on each quaternionic pixel value x as:

$$lReLU_q(x) = lReLU_r(x_a) + lReLU_r(x_b)\boldsymbol{i} + lReLU_r(x_c)\boldsymbol{j} + lReLU_r(x_d)\boldsymbol{k} \quad (15)$$

where $lReLU_r$ is the well-known real-valued leaky ReLU function and we assume $x = x_a + x_b\boldsymbol{i} + x_c\boldsymbol{j} + x_d\boldsymbol{k}$. The generator implements a mapping $\mathbb{H}^{H \times W} \rightarrow [0, 1]^{H \times W}$, from a quaternion-valued image to a real image. All intermediate layers map quaternion-valued feature maps again to quaternion-valued feature maps, save for the final activation. We define the final activation simply as the sum:

$$qsum(x) = x_a + x_b + x_c + x_d. \quad (16)$$

which ensures a real-valued output.

The discriminator is constructed as a cascade of strided quaternionic convolutions, with strides and size identical to those used for the generator encoder. It

implements a mapping $\mathbb{H}^{H\times W} \to [0,1]$, where the output represents the degree in which the network believes that the input is fake or genuine. Inputs to the discriminator are constructed as concatenations of color inscription images to the estimated target. In particular, we map *Detection estimate* $\to$ *real part*, *Red* $\to \boldsymbol{i}$, *Green* $\to \boldsymbol{j}$, *Blue* $\to \boldsymbol{k}$. As the output is real while the input is quaternionic, in the final layer we use the activation of eq. (16), before applying a sigmoid function on top of it. The discriminator is made up of 6 quaternionic convolutional layers. Output number of channels equal to $16, 32, 64, 64, 128, 1$ respectively for the 6 convolutional layers.

Note also that the setup of the generator and discirminator is such that inputs and outputs can be of variable size. Indeed, the generator is a fully convolutional network, with parameters and layers that are independent of input and feature map size. The discriminator leads to feature maps that are reduced to a single probability value, again regardless of the input image and annotation dimensions.

6 Experimental results

6.1 Dataset

The dataset is comprised of a total of 67 images containing inscriptions written in Greek, and found in Byzantine churches and monasteries in the region of Epirus, located in Northwestern Greece [9, 21]. Our inscriptions are donor's inscriptions, typically made up of a few lines of text and containing information about who donated funds and other resources required to build the monument where the inscription is located. The photographed images were captured with a Samsung GT-I9505 and a Nikon Coolpix L810 camera. All images were then resized so as their width was at most 1024 pixels, keeping their aspect ratio fixed. We have chosen to partition the set to a training and test set according to a 80%/20% rule, which resulted to training and test sets of 55 and 12 images respectively.

6.2 Experiments

Concerning training, we have used the Adam optimizer with parameters $\beta_1 = 0.5$, $\beta_2 = 0.999$. No data augmentation is used. The trade-off parameter λ was set to 10 and base learning rates were set to 10^{-4} for the discriminator and 5×10^{-4} for the generator. Furthermore, a learning rate scheduling strategy was used, where learning rate is divided by 10 for both networks if test binary cross-entropy deteriorates continuously for 2 consecutive epochs. Batch size was set to 1, as our model was setup to accept inputs of variant size.

We have used two evaluation measures: a) binary cross-entropy (BCE) of the test images and b) Intersection over Union (IoU). Test BCE is applied in an analogous manner to the corresponding loss component discussed in section 5, and effectively tests for correct per-pixel binary classification. The IoU measure is applied after computing a binarized version of the estimate detection map, with a threshold of 0.5 (Pascal VOC challenge [3]). Subsequently, IoU is computed between this binarized estimate and the ground truth.

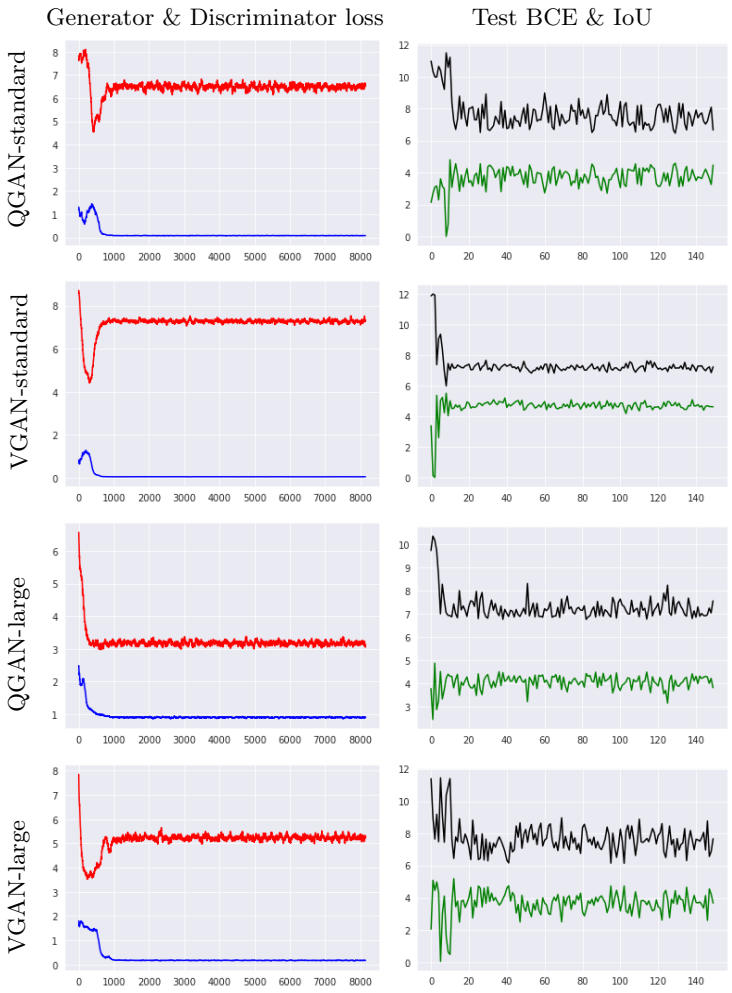

Fig. 3. Generator loss, Discriminator loss, Test BCE loss and IoU score plots for all models tested in this work. From top row to bottom, we show results for QGAN-standard, VGAN-standard, QGAN-large, VGAN-large. Left column shows Generator and Discriminator loss (red and blue respectively, lower is better for both), and right column shows test BCE and IoU (black and green respectively. Lower BCE is better, higher IoU is better). Generator and Discriminator losses are smoothed with a 100-point uniform convolution kernel and plotted per iteration, test BCE and IoU are plotted per epoch. IoU score is shown multiplied 10× for better visualization.

In figure 3, we show plots for the generator and discriminator loss calculated per training iteration, and test BCE loss and IoU calculated as an average over test images and at the end of each epoch. QGAN and VGAN are compared, as well as two considered model sizes. Standard size corresponds to the model

described previously in section 5. Large size corresponds to QGAN and VGAN models that have double the number of channels per convolutional or deconvolutional layer.

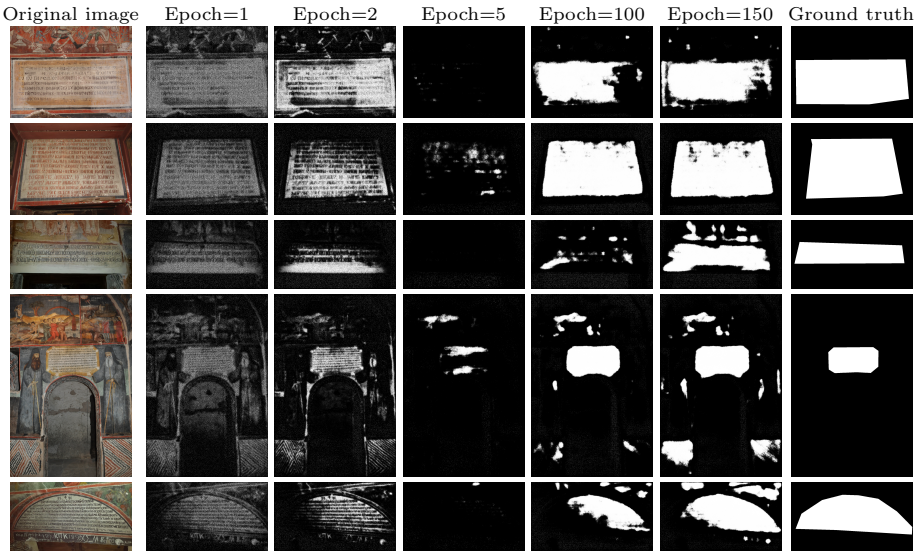

Fig. 4. Sample results of proposed model on test images.

We show results for test images as a function of current epoch training in figure 4.

We compare each Quaternionic GAN model with its vanilla (non- quaternionic) counterpart, by considering a network with the same amount of neurons. For each quaternionic neuron of the QGAN, we need to create four neurons for the corresponding VGAN, due to the isomorphism between $\mathbb{H}$ and $\Re^4$. As shown before [17], computation of the quaternionic (Hamilton) product and consequently quaternionic convolution requires considerably less storage. Judging from the results shown in fig. 3 and table 1, we can conclude that in all cases performance of the proposed QGAN is comparable with its corresponding non-quaternionic model, and definitely with scores on the same order of magnitude. IoU scores seems somewhat worse, though BCE results are more inconclusive, with QGAN faring slightly better than VGAN with respect to the "Standard" model size. What is definitely noteworthy though, is that QGAN is a considerably less expensive network (in table 2 we show the number of total network parameters for each version of the QGANs and VGANs considered). The number of weights, translated in practice in required storage, is only 25% of the non-quaternionic versions. This means that the proposed QGAN can achieve similar results with the standard GAN, using four times less parameters.

Table 1. Numerical results for two variants of the proposed model (QGAN) versus its non-quaternionic counterpart with the same number of neurons (VGAN). Test BCE figures (lower is better) are shown and corresponding IoU scores in parenthesis (higher is better).

Model / Network type	Standard	Large
Quaternionic GAN	6.54(45.4%)	6.91(44.9%)
Vanilla GAN	7.4(51.9%)	6.45(52.0%)

Table 2. Comparative table of model sizes, measured in numbers of trainable weights. Number of quaternionic and real weights are shown respectively. In parenthesis, the number of equivalent real weights is shown, in order to ease storage size requirements comparison for the two variants.

Model / Network type	Standard	Large
Quaternionic GAN	$381,426$ $(1,525,704)$	$1,516,514$ $(6,066,056)$
Vanilla GAN	$6,053,826$	$24,166,274$

7 Conclusion and Future work

We have presented a new variant of Generative Adversarial Networks that uses quaternion-valued neurons and weights, as well as suitable quaternionic variants of convolutional and deconvolutional layers. The proposed model is a conditional GAN, with the generator accepting a color input image and outputing a detection heatmap. We have applied the new model on the task of inscription detection, where we have used a set of byzantine monument text inscriptions as our targets. Quaternion-valued networks such as the proposed one can inherently deal with representing color intercorrelation. The inscriptions themselves are not characterized by color variance; however, the elements that are not part of the inscription very often do (murals, paintings). The proposed network showed that it can be as effective as a real-valued GAN, while being much less expensive in terms of model size. This can be a very important factor, especially in use cases where the resource budget is very constrained (e.g. neural networks running on mobile phones, etc.).

As future work, we plan to conduct more extensive sets of experiments, testing multiple architectures as well as other GAN variants, or experiment with spatially-constrained adjustments to the loss function [15]. Alternative features may also be considered, such as QFT-based cues [6]. Extensive augmentation is another technique that we can plan to explore, especially using perspective transforms to simulate alternate viewpoints, or experimenting with learning-based augmentation [1].

Acknowledgements

We would like to thank Dr. Christos Stavrakos, Dr. Katerina Kontopanagou, Dr. Fanny Lyttari and Ioannis Theodorakopoulos for supplying us with the Byzantine inscription images used for our experiments.

We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan XP GPU used for this research.

This research has been partially co-financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call OPEN INNOVATION IN CULTURE, project *Bessarion* (T6YBΠ-00214).

References

1. Dimitrakopoulos, P., Sfikas, G., Nikou, C.: Ising-gan: Annotated data augmentation with a spatially constrained generative adversarial network. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI). pp. 1600–1603. IEEE (2020)
2. Ell, T.A., Sangwine, S.J.: Hypercomplex fourier transforms of color images. IEEE Transactions on image processing **16**(1), 22–35 (2007)
3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. International Journal of Computer Vision **88**, 303–338 (2010)
4. Giotis, A.P., Sfikas, G., Gatos, B., Nikou, C.: A survey of document image word spotting techniques. Pattern Recognition **68**, 310 – 332 (2017)
5. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems (NIPS). pp. 2672–2680 (2014)
6. Hui, W., Xiao-Hui, W., Yue, Z., Jie, Y.: Color texture segmentation using quaternion-gabor filters. In: 2006 International Conference on Image Processing. pp. 745–748. IEEE (2006)
7. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. arXiv preprint arXiv:1611.07004 (2016)
8. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. International Journal of Computer Vision **116**(1), 1–20 (2016)
9. Kordatos, E., Exarchos, D., Stavrakos, C., Moropoulou, A., Matikas, T.: Infrared thermographic inspection of murals and characterization of degradation in historic monuments. Construction and Building Materials **48**, 1261–1265 (2013)
10. Leung, H., Haykin, S.: The complex backpropagation algorithm. IEEE Transactions on signal processing **39**(9), 2101–2104 (1991)
11. Liao, M., Shi, B., Bai, X.: Textboxes++: A single-shot oriented scene text detector. IEEE Transactions on Image Processing **27**(8), 36763690 (Aug 2018)
12. Liao, M., Zhu, Z., Shi, B., song Xia, G., Bai, X.: Rotation-sensitive regression for oriented scene text detection (2018)
13. Lucic, M., Kurach, K., Michalski, M., Gelly, S., Bousquet, O.: Are GANs created equal? a large-scale study. In: Advances in neural information processing systems (NIPS). pp. 700–709 (2018)

14. Nitta, T.: A quaternary version of the back-propagation algorithm. In: Proceedings of ICNN'95-International Conference on Neural Networks. vol. 5, pp. 2753–2756. IEEE (1995)
15. Papadimitriou, K., Sfikas, G., Nikou, C.: Tomographic image reconstruction with a spatially varying gamma mixture prior. Journal of Mathematical Imaging and Vision **60**(8), 1355–1365 (2018)
16. Parcollet, T., Morchid, M., Linarès, G.: Quaternion convolutional neural networks for heterogeneous image processing. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 8514–8518. IEEE (2019)
17. Parcollet, T., Morchid, M., Linares, G.: A survey of quaternion neural networks. Artificial Intelligence Review **53**(4), 2957–2982 (2020)
18. 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)
19. Raisi, Z., Naiel, M.A., Fieguth, P., Wardell, S., Zelek, J.: Text detection and recognition in the wild: A review (2020)
20. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks (2016)
21. Rhoby, A.: Text as art? byzantine inscriptions and their display. Writing Matters: Presenting and Perceiving Monumental Inscriptions in Antiquity and the Middle Ages. Berlin: de Gruyter pp. 265–83 (2017)
22. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
23. Su, F., Ding, W., Wang, L., Shan, S., Xu, H.: Text proposals based on windowed maximally stable extremal region for scene text detection. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 01, pp. 376–381 (2017)
24. Yao, C., Bai, X., Sang, N., Zhou, X., Zhou, S., Cao, Z.: Scene text detection via holistic, multi-channel prediction (2016)
25. Ye, Q., Doermann, D.: Text detection and recognition in imagery: A survey. IEEE Transactions on Pattern Analysis & Machine Intelligence **37**(07), 1480–1500 (2015)
26. 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)