Document Binarization with Quaternionic Double Discriminator Generative Adversarial Network

Giorgos Sfikas[1](✉), George Retsinas[2], and Basilis Gatos[3]

[1] Department of Surveying and Geoinformatics Engineering, School of Engineering, University of West Attica, Athens, Greece
gsfikas@uniwa.gr

[2] School of Electrical and Computer Engineering, National Technical University of Athens, Athens, Greece
gretsinas@central.ntua.gr

[3] Computational Intelligence Laboratory, Institute of Informatics and Telecommunications, National Center for Scientific Research "Demokritos", Athens, Greece
bgat@iit.demokritos.gr

Abstract. Quaternionic networks have emerged as a lightweight alternative to standard neural networks. We propose using a Quaternionic conditional Generalized Adversarial Network adapted to document image binarization. A double discriminator ensures that the output is consistent over a coarse and a finer level of resolution, while the generator is tasked with producing the binarized document. We achieve excellent binarization results, while our network is significantly smaller (4x smaller) than its real-valued counterpart.

Keywords: Binarization · Quaternions · Generative Adversarial Nets

1 Introduction and Related Work

Document image binarization is the task of converting a scanned document image into a binary segmentation. One segment corresponds to the content of the document, while the other segment must cover non-textual components. Binarization is usually desired in order to reduce storage space for large document collections, save communication bandwidth, prepare the document for a subsequent document processing step, or simply enhance the document for better readability. Binarization can be especially challenging when it comes to handling scanned historical manuscripts, as its efficiency is affected by various types of degradation. Poor preservation, bleed-through, faded ink, stains, are some the most common "culprits" that make successful binarization, as well as other processes of document analysis, even more difficult. When it comes to modern documents, another set of problems is related to contemporary picture capturing conditions, e.g. non-uniform lighting due to use of strobe flash [13].

M. Coustaty and A. Fornés (Eds.): ICDAR 2023 Workshops, LNCS 14194, pp. 272–284, 2023.
https://doi.org/10.1007/978-3-031-41501-2_19

Classical binarization techniques were relying on treating the problem as that of choosing a single, global threshold of intensity that would be used to separate background from foreground segments. A threshold is selected automatically according to optimizing an objective that is based on the image intensity statistics [19]. Global thresholding is definitely suboptimal, as its underlying premise is false: No single threshold is applicable for all pixels in an image, and image statistics are in general non-stationary. The immediate extension of global thresholding has been adaptive thresholding, or choosing a different threshold for each different area of patch of the image [8,20,25]. Another group of methods has focused on fitting the image to a more elaborate model of image formation. A Total Variation framework is used in [11], where a data fidelity is traded off a regularization term, which encodes a prior belief that both segments must be spatially consistent, all the while preserving separating edges. Postprocessing is not uncommon, and can be used to rectify the output of an initial processing phase. Such tools may include morphological image processing operations (opening, closing, erosion, dilation, etc.) or non-local filtering [11].

After the starting gun for the comeback of neural networks and deep learning has been fired with Alexnet and other developments around 2014, it has not been long before document image processing techniques were also flooded with deep model solutions. Regarding binarization in particular, neural networks are essentially treated as complex, non-linear filters that are to be learned from data. To this end, fully convolutional networks (FCNs) have been the go-to solution in most cases [2,4,31,33,34]; exceptions include the recurrent architecture proposed in [34], where Grid Long-Short Term Memory units are used. In general however, convolutional architectures provide a very useful inductive bias when processing images. Although more recent competitors such as transformers [23] claim to have rendered convolutions and recurrent units next to unnecessary [32], convolutions especially seem to remain always ubiquitous in cutting-edge vision models in general. In [17], a "morphological" neural network learns a set of morphological operations over an input image; these operations include the weights of a layer that combines dilation and erosion operations, as well as the parameters of the morphological structuring element.

Recently, authors have set forward considerations other than processing accuracy [13]. Model size is another factor that can be very important, especially when it comes to applying vision techniques on embedded systems, or in general machines that are heavily resource-constrained. Methods that involve pruning weights or neurons, or whole blocks of model components is one strategy that may result in a neural network that is as efficent, or almost as efficient, as the non-pruned network [5,15,24] The advantage in this case is that due to the smaller size of total parameters, we have in this sense a smaller network and less storage requirements, as well as potentially speedier inference. Recent work in network pruning methods shows that it is not rare to attain good model compression rates at little loss of efficiency (e.g. [5]).

Hypercomplex architectures constitute a line of work within the research direction that aims for smaller models [21]. This is a type of neural network architectures that is based on challenging the *de facto* choice of using real-valued representations in all aspects of neural networks. In hypercomplex networks, we have inputs, intermediate outputs, weights, biases and activations that are defined on hypercomplex number domains. Hypercomplex numbers in a sense generalize the notion of complex numbers, where each number is defined as a composition of a real and an imaginary part, with an imaginary part that can itself comprise multiple, independent dimensions and multiple, orthogonal imaginary axes. The most well-studied hypercomplex algebra is that of quaternions. Aside from the major application of quaternions involving representations of spatial rotations [10], in computer science quaternions have been used in the domains of signal processing, digital image processing and computer vision [1,6,30]. An application, for which the theoretical prerequisite has been laid down in the 90s [18], is that of quaternion neural networks. It may seem that superficially there is no gain in representing groups of 4 as quaternions on the basis of a trivial bijection between quaternions $\mathbb{H}$ and four-dimensional vectors in $\mathbb{R}^4$; however, due to using quaternion operations – and specifically, quaternion multiplication – the network under the hood uses significantly less parameters than a real-valued model of corresponding size [21].

In this work, we propose a lightweight, quaternion-based architecture over a state-of-the-art backbone that is based on a conditional Generative Adversarial Network geared for the task of document image binarization. To this end, the network includes two key components: namely, i. a double discriminator that is intended to check that the generated binarization over a coarse and a fine resolution. ii. focal loss, which acts to mitigate the detrimentary effect of having imbalanced background and foreground classes. All components are defined in terms of quaternionic operations, including fully-connected layers, convolutions, and activations. We achieve results with little or minimal loss in accuracy over tests in the DIBCO 2017 dataset [22], all the while using a network with a total size that is 4 times smaller than the real-valued model.

The remainder of this paper is structured as follows. In Sect. 2 we briefly review the theoretical requirements concerning quaternions. In Sect. 3 we present the proposed model, and in Sect. 4 we present numerical evaluation results. We close with our concluding remarks in Sect. 5.

2 Quaternion Neural Networks

2.1 Preliminaries

In this section, we shall provide a brief introduction in the theoretical preliminaries regarding quaternions and hypercomplex numbers in general. Sets of hypercomplex numbers are forms of mathematical structures that are comprised of numbers of "higher-dimensionality". Dimensionality in this sense can be understood as the dimension of the linear space with which each corresponding set is isomorphic. For example, the set $\mathbb{H}$ of quaternions is isomorphic to the set of 4-dimensional vectors of $\mathbb{R}^4$ (disregarding for the time being other forms of algebraic structure), as we can trivially define a bijection between the two sets.

This is straightforward from the definition of a quaternion, which is as follows. Any $q \in \mathbb{H}$ can be written as:

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

where a, b, c, d are real numbers, and $\boldsymbol{i}, \boldsymbol{j}, \boldsymbol{k}$ are so-called imaginary units which correspond to an equal number of imaginary axes. The "a" component corresponds to the real axis. From this definition, we can see that quaternions are a generalization of the set of real numbers $\mathbb{R}$ and the set of complex numbers $\mathbb{C}$. Indeed, for $b = c = d = 0$ we obtain a quaternion that is also a real number; for $c = d = 0$ we obtain a complex number.

With respect to algebraic structure, we can endow $\mathbb{H}$ with summation and product operations so that they form a division algebra or skew-field [7]. This means quaternions $\mathbb{H}$ adhere to all the properties of an algebraic field, such as e.g. the field $\mathbb{R}$, with the exception of being commutative with respect to multiplication. So, in general we have $pq \neq qp$ for $p, q \in \mathbb{H}$. Regarding the definitions of addition and multiplication, the former is simply a sum of corresponding real or imaginary components. In particular,

$$p + q = (a_p + a_q) + (b_p + b_q)\boldsymbol{i} + (c_p + c_q)\boldsymbol{j} + (d_p + d_q)\boldsymbol{k}, \tag{2}$$

where we use $p = a_p + b_p\boldsymbol{i} + c_p\boldsymbol{j} + d_p\boldsymbol{k}$ and $q = a_q + b_q\boldsymbol{i} + c_q\boldsymbol{j} + d_q\boldsymbol{k}$. Multiplication over $\mathbb{H}$ is more complex; we can break down its definition by defining first a multiplication rule between pairs of real and imaginary units. The identity 1 of course leaves any q intact by definition, $1q = q$, and for the imaginary units we have:

$$i^2 = j^2 = k^2 = -1, ij = -ji = k, jk = -kj = i, ki = -ik = j, ijk = -1. \tag{3}$$

From the above relation we immediately can see that the square root of -1 does not correspond only to $\pm\boldsymbol{i}$ as in the set of complex numbers $\mathbb{C}$, but to the further two imaginary units as well; furthermore, the equation $\mu^2 + 1 = 0$ in fact possesses an infinite number of solutions in $\mathbb{H}$. By combining the Eqs. 2 and 3 with the distributive property, we readily obtain the multiplication rule for quaternions p, q:

$$\begin{aligned} pq =& (a_p a_q - b_p b_q - c_p c_q - d_p d_q) + \\ & (a_p b_q + b_p a_q + c_p d_q - d_p c_q)\boldsymbol{i} + \\ & (a_p c_q - b_p d_q + c_p a_q + d_p b_q)\boldsymbol{j} + \\ & (a_p d_q + b_p c_q - c_p b_q + d_p a_q)\boldsymbol{k}. \end{aligned} \tag{4}$$

This rule is also know as a *Hamilton* product [21]. It is especially important regarding the application of quaternions to neural networks, as one of the major differences between "standard" neural network layers and quaternionic layers is that multiplication in all cases is performed according to Eq. 4.

Another way of representing quaternions, is by writing them as a sum of two components. In turn, this can be done in (at least) two ways. One way is to consider the real part to be a single component $S(q)$, and the rest as another component $V(q)$. In this manner, one part corresponds to the real axis, and the other part corresponds to the imaginary axes collectively.

$$q = S(q) + V(q), \tag{5}$$

where $S(q) = a$ and $V(q) = b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$. Another way to represent quaternions is by the Caley-Dickson construction. This amounts to writing $q \in \mathbb{H}$ as a sum of a real and imaginary part, like members of $\mathbb{C}$:

$$q = \alpha + \beta\boldsymbol{k}, \tag{6}$$

where $\boldsymbol{k}$ is the imaginary unit we defined previously, and $\alpha, \beta \in \mathbb{C}$ (generalizing the analogous construction for $\mathbb{C}$, where $\alpha, \beta \in \mathbb{R}$). Assuming $\alpha = \gamma + \delta\boldsymbol{i}$ and $\beta = \epsilon + \zeta\boldsymbol{i}$, it is straightforward again to combine with the imaginary unit multiplication rule of Eq. 3 and the distributive property in $\mathbb{H}$, to obtain our initial definition in Eq. 1.

Furthermore, many properties and notions that are well-known from $\mathbb{R}$ or $\mathbb{C}$ are inherited to, or are generalized gracefully to elements of $\mathbb{H}$. For example, we define a length or magnitude of a quaternion as: $|q| = \sqrt{q\overline{q}} = \sqrt{\overline{q}q} = \sqrt{a^2 + b^2 + c^2 + d^2}$, where $\overline{q}$ is the conjugate of q, defined as $\overline{q} = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}$. Quaternions with a zero real part are called pure quaternions, and quaternions with unitary length $|q| = 1$ are called unit quaternions. The Taylor series is a very useful tool that generalizes to $\mathbb{H}$, namely as $e^p = \sum_{n=0}^{\infty} \frac{p^n}{n!}$. Given a quaternion p that is both unit and pure, we also obtain a generalization of Euler's identity, $e^{p\theta} = cos\theta + p \sin\theta$. It is trivial to see that for $p = \boldsymbol{i}$ we get Euler's identity for complex numbers. Quaternions can also be written in polar form: $q = |q|e^{\mu\theta}$, with $\theta \in \mathbb{R}$, and $p \in \mathbb{H}$ again unit and pure. Quaternion p and real angle θ are referred to as the eigenaxis and eigenangle [1]. The eigenaxis and eigenangle can be computed as: $\mu = V(q)/|V(q)|, \theta = \tan^{-1}(|V(q)|/S(q))$. For pure q, hence $S(q) = 0$, we have $\theta = \pi/2$.

Extensions of convolution are also of special importance to applications to quaternionic neural networks. As quaternion multiplication is non-commutative, it is perhaps unsurprising that convolution of signals f and g is also non-commutative, $f * g \neq g * f$. Hence, different variants of convolution can be employed, depending on whether the kernel multiplies the signal from the left or the right; in $2D$ especially, a bi-convolution operation can also be defined, which has one part of the kernel multiplying the input from the left, and the other part from the right. For all intents and purposes within the scope of the current application, we treat these variants as equivalent; we shall use a left-side convolution by convention:

$$(w * f)(k, l) = \int_{\Omega_y} \int_{\Omega_x} w(x, y) f(k - x, l - y) dx dy \tag{7}$$

As a sidenote, a set of other interesting properties of quaternions emerge once we consider matrices with quaternionic values. Properties that are well-known from complex matrix algebra may generalize naturally to quaternions, while others may diverge immensely from our experience on $\mathbb{R}$ or even $\mathbb{C}$. For example, square quaternionic matrices do have sets of eigenvalues and eigenvectors, in the sense of vectors that solve the equation $Ax = \lambda x$, for $A \in \mathbb{H}^{N \times N}$, $x \in \mathbb{H}^N$, $\lambda \in \mathbb{H}$. However, since in general $\lambda x \neq x\lambda$ due to non-commutativity of multiplication in $\mathbb{H}$, the problems $Ax = \lambda x$ are $Ax = x\lambda$ are different. The two sets of eigenvalues are distinct, and are referred to as *left eigenvalues* and *right eigenvalues* respectively. Very little is known regarding connections between the two sets [16,36].

2.2 Quaternion Layers

Conversion of "standard" linear transformations (in the sense of assuming real values on inputs, outputs and transformation parameters) is the "workhorse" behind converting real-valued networks to quaternionic ones. Recall that in general a feed-forward neural network can be written as a composition of ℓ layers, which are in turn defined as compositions of a linear and a non-linear part. Formally we write:

$$f[x, \theta] = f^{\ell} \circ f^{\ell-1} \circ \cdots \circ f^{1}[x, \theta], \tag{8}$$

where $f[x, \theta]$ represents the NN, which takes an input x conditioned on parameters θ and outputs a vector y. All components are made up of quaternionic variates, $x \in \mathbb{H}^{d_x}$, $\theta \in \mathbb{H}^{d_\theta}$, $y \in \mathbb{H}^{d_y}$, where d_x, d_θ, d_y represent input, parameter and output dimensionalities. Note that in practical terms, assuming that we have a network consisting of d_θ quaternionic parameters, they will take up as much space as $4d_\theta$, since each quaternion is intrinsically 4-dimensional. Layers are represented as $f^1, \cdots, f^{\ell}$ in the above formulation. In general, each layer consists of a linearity or linear component $f_L(\dot{})$ and a non-linearity or activation $f_{NL}(\dot{})$.

Quaternion Linearities. The linear component is written as:

$$g = Wf + b, \tag{9}$$

where $g \in \mathbb{H}^M$, $f \in \mathbb{H}^N$ represent layer outputs and inputs, and W, b are the weights and biases of the layer, which are of course part of the full set of parameters θ. W is a quaternionic matrix $\mathbb{H}^{M \times N}$ and b is a quaternionic vector $\mathbb{H}^M$. We can rewrite this relation as:

$$g^i = \sum_{j=1}^{N} w^{ij} f^j + b^i, \tag{10}$$

where additions and multiplications follow the rules for quaternions (cf. Sect. 2). Multiplication between w^{ij} and f^j is effectively a Hamilton product (Eq. 4). We then use the fact that we can rewrite Eq. 4 as a matrix-vector product, where one of the quaternions (here, the weight component w_{ij}) is rewritten as a 4×4 (real) matrix:

$$\begin{bmatrix} g_a \\ g_b \\ g_c \\ g_d \end{bmatrix} = \begin{bmatrix} w_a & -w_b & -w_c & -w_d \\ w_b & w_a & -w_d & w_c \\ w_c & w_d & w_a & -w_b \\ w_d & -w_c & w_b & w_a \end{bmatrix} \begin{bmatrix} f_a \\ f_b \\ f_c \\ f_d \end{bmatrix}, \tag{11}$$

The key observation now is that we can combine Eqs. 9 and 11 and use the trivial bijection between $\mathbb{R}^4$ and $\mathbb{H}$, i.e. $(a, b, c, d)^T \rightarrow (a + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k})$, in order to write Eq. 9 in a block-matrix form as follows:

$$\begin{bmatrix} \boldsymbol{g}^1 \\ \boldsymbol{g}^2 \\ \cdots \\ \boldsymbol{g}^M \end{bmatrix} = \begin{bmatrix} \boldsymbol{w}^{11} & \dots & \boldsymbol{w}^{1N} \\ \vdots & \ddots & \vdots \\ \boldsymbol{w}^{M1} & \dots & \boldsymbol{w}^{MN} \end{bmatrix} \begin{bmatrix} \boldsymbol{f}^1 \\ \boldsymbol{f}^2 \\ \cdots \\ \boldsymbol{g}^N \end{bmatrix} + \begin{bmatrix} \boldsymbol{b}^1 \\ \boldsymbol{b}^2 \\ \cdots \\ \boldsymbol{b}^N \end{bmatrix}. \tag{12}$$

In the above equation each boldface element represents a 4×1 vector (on the vectors) or a 4×4 submatrix (on the matrix). Hence, we are dealing with dimensions equal to $4d_g$, $4d_f$ for the input and output vectors. However, due to the bijective relation between the two equation forms Eq. 11 and 12, *the multiplying matrix only has $4d_g d_f$ independent parameters.* A real matrix construction would have $4 * 4d_g d_f$ independent parameters, i.e. equal to the number of all matrix elements. Thus, we have four-fold saving in number of parameters (similar discussions concerning why quaternion layers lead to extensive parameter savings can be found in e.g. [21] or [28, Section 3])

Convolutions can be interpreted as a constrained version of the fully connected layer, where extensive parameter sharing is employed in the form of the convolution kernel. It is well-known that convolutions, as linear operations, can be written in a matrix-vector form as Töplitz matrices [9]. Hence, the entirety of the aformentioned analysis also applies in their case. Similar considerations hold for transpose convolutions or deconvolutions.

Quaternion non-linearities. Regarding activation functions, we formally require a mapping from $\mathbb{H}$ to $\mathbb{H}$. In practice, so-called split activation functions are usually employed, where simply real-valued activations are used over each of the quaternionic (real/imaginary) componenets separately.

3 Proposed Model

The proposed architecture is based on a Quaternionic conditional Generative Adversarial Network, comprised of a total of three composing networks: A generator network, tasked with producing a binarization given the original input image; a global discriminator network, tasked with discerning between produced binarization that are unlikely to be artificial and those that are; a local discriminator network, which acts similarly as the global discriminator but on the level of small-sized patches (32×32 pixels). In this manner, the binarization estimate can be evaluated by the networks in two resolution scales: a coarse one

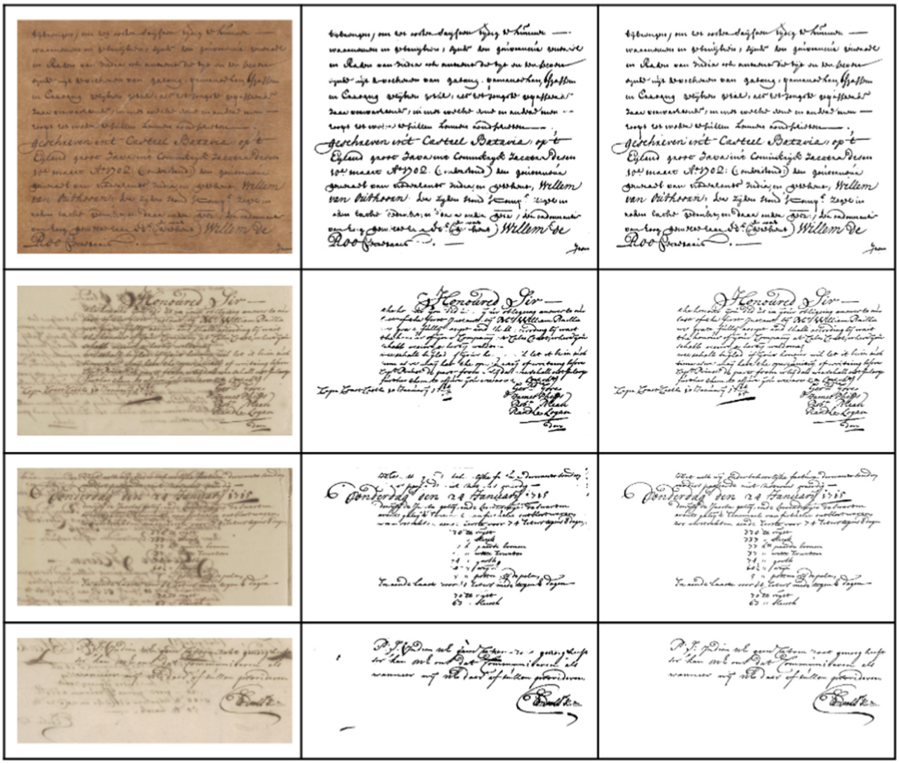

Fig. 1. Visual results using the proposed Quaternionic Double-Discriminator GAN model. Lines correspond to selected images from the DIBCO 2017 test set. Columns correspond to: original image, binarization result, ground truth.

that corresponds to the global discriminator, and a finer one that corresponds to the local discriminator. The generator is structured as a U-Net, with skip connections between corresponding layers of the same size. Global and local discriminators are defined as in [4], with the exception of adding 1×1 convolution layers when it is necessary to change input or output channel depth to a multiple of 4. Training loss is defined as a weighted average of Binary cross-entropy losses for the three composing networks. Formally we have:

$$L = \mu(L_{global} + \sigma L_{local}) + \lambda L_{gen}. \tag{13}$$

We set loss aggregation hyperparameters to $\mu = 0.5, \sigma = 5, \lambda = 75$, as suggested in [4]. Concerning imbalances in the number of background and foreground elements, one major strategy is using resampling in order to prioritize classes that are initially under-represented, by sampling more augmented samples from low-volume classes. The other major strategy is to tweak the loss function, so that under-represented classes are artifically assigned a larger loss. In this manner, training can be implicity manipulated towards an optimum that classifies small classes as well the larger ones. The latter rationale is followed by Focal loss [12], employed for training the current model.

4 Experiments

We have used the DIBCO 2017 dataset to test the proposed quaternionic GAN. All grayscale inputs are augmented with three extra zero channels so as to be able to be cast as quaternions. Each image is broken into 256 × 256-sized patches with a stride equal to 128. As in [4], all the images from previous DIBCO competitions together with DIBCO 2018 have been used to train the model. In Table 3 we review results of the proposed model against other state-of-the-art methods, using the following metrics: F-measure (F-m), pseudo F-measure (pseudo F-m), distance reciprocal distortion metric (DRD), peak Signal-to-Noise Ratio (PSNR). In Tables 1 and 2, a more detailed report of results is presented. It compares our model versus its non-quaternionic counterpart [4], and furthermore we report extra metrics – Recall, Precision, PseudoRecall and PseudoPrecision. This is done for all of the DIBCO 2017 test images separately. Comparing the

Table 1. Binarization numerical results using the proposed Quaternion Double-Discriminator GAN. Each line corresponds to a different DIBCO 2017 test image.

	F-m	Pseudo F-m	PSNR	DRD	Recall	Precision	PseudoRecall	PseudoPrecision
1	71.85	68.02	12.08	10.52	83.82	62.87	85.05	56.67
2	84.81	82.54	15.60	6.81	94.22	77.59	96.77	74.21
3	75.56	71.83	13.46	9.77	89.28	65.5	91.09	59.3
4	71.88	68.32	14.11	0.87	83.28	63.23	84.89	57.16
5	74.84	70.68	16.08	11.54	95.94	61.35	96.78	55.67
6	94.4	95.33	15.96	2.22	96.1	92.76	99.7	91.33
7	94.12	94.4	15.99	2.35	96.83	91.55	99.56	89.74
8	92.79	95.1	19.43	2.64	92.97	92.61	99.09	91.42
9	86.55	84.1	14.95	4.52	96.35	78.57	96.13	74.74
10	88.6	86.19	14.38	4.03	99.16	80.08	99.62	75.95
11	92.84	91.72	16.79	3.73	99.57	86.96	99.76	84.87
12	84.59	83.93	14.86	7.69	96.83	75.09	99.1	72.79
13	66.2	64.55	11.5	25.38	97.65	50.07	99.65	47.74
14	87.51	87.29	16.78	5.75	96.98	79.73	99.26	77.9
15	93.31	93.42	16.77	2.72	97.04	89.86	98.99	88.45
16	87.49	84.86	18.09	5.23	99.44	78.11	99.77	73.83
17	77.4	72.64	15.37	7.68	95.3	65.16	95.72	58.53
18	84.4	87.75	14.95	7.05	88.18	80.93	98.21	79.3
19	91.67	92.57	18.98	3.1	93.98	89.46	99.55	86.51
20	91.42	93.92	16.95	3.29	92.82	90.06	99.73	88.75
Avg	84.61	83.45	15.60	6.81	94.22	77.59	96.77	74.21

Table 2. Binarization numerical results using a non-quaternionic Double-Discriminator GAN (baseline model, cf. [4]). Each line corresponds to a different DIBCO 2017 test image. Compared to the proposed model, similar results are achieved, albeit at a much heavier computational burden.

	F-m	Pseudo F-m	PSNR	DRD	Recall	Precision	PseudoRecall	PseudoPrecision
1	72.25	68.25	12.12	10.36	84.68	63	85.77	56.68
2	85.79	83.29	14.86	5.92	96.46	77.24	97.26	72.83
3	76.28	72.16	13.45	9.8	93.2	64.56	94.89	58.22
4	72.25	68.34	13.94	11.44	87.96	61.3	89.57	55.24
5	73.55	69.74	15.82	12.56	95.27	59.89	96.28	54.67
6	93.98	94.75	15.62	2.52	96.29	91.78	99.7	90.26
7	93.79	94.03	15.74	2.63	96.95	90.83	99.68	9.05
8	93.23	95.05	19.68	2.51	94.06	92.42	99.19	91.24
9	86.43	83.96	14.86	4.65	97.17	77.82	97.31	73.83
10	88.6	86.23	14.38	4.04	99.08	80.12	99.6	76.03
11	92.9	91.85	16.84	3.68	99.49	87.13	99.75	85.11
12	85.13	84.48	15.04	7.3	97.01	75.84	99.33	73.5
13	66.6	64.94	11.59	24.7	997.48	50.58	99.63	48.17
14	88.7	88.5	17.26	5.07	97.18	81.57	99.44	79.7
15	93.2	693.3	16.73	2.7	97.27	89.57	99.21	88.05
16	88.8	86.39	18.64	4.42	99.27	80.33	99.72	76.2
17	75.76	71.46	15.28	7.86	88.87	66.03	89.49	59.48
18	85.37	88.75	15.25	6.49	88.79	82.2	98.91	80.48
19	92.68	93.92	19.59	2.5	93.87	91.53	99.47	88.95
20	92.65	95.17	17.67	2.6	92.92	92.37	99.71	91.03
Avg	84.9	83.72	15.71	6.69	94.66	77.80	97.19	74.43

Table 3. Comparison of state-of-the-art document binarization methods, using DIBCO 2017 test set accuracy as a benchmark. Lines denoted as "Comp #x" refer to winners of the corresponding competition [22]. DD-GAN-x refers to a baseline, real-valued double discriminator model. The results under DD-GAN-1 are the figures reported in the original publication, [4], while DD-GAN-2 corresponds to the figures we computed after running the authors' implementation [3], without any parameter finetuning. Quaternion DD-GAN refers to the proposed method, which achieves good results while being very economical in terms of network size (4× smaller compared to DD-GAN).

	F-measure	Pseudo F-measure	PSNR	DRD
Comp #1 (U-Net)	91.04	92.86	18.28	3.4
Comp #1 (FCN-VGG)	89.67	91.03	17.58	4.35
Comp #3 (Deep SN)	89.42	91.52	17.61	3.56
Otsu	77.73	77.89	13.85	15.54
Sauvola	77.1	77.89	14.25	8.85
DD-GAN-1	90.98	92.85	17.6	3.34
DD-GAN-2	84.9	83.72	15.71	6.69
Quaternion DD-GAN	84.61	83.45	15.6	6.81

two tables, we can conclude that in may cases we can observe a slight loss in performance, though for the most part losses are insignificant. On the other hand, the proposed model is 4× smaller (cf. Subsect. 2.2) than the real-valued state-of-the-art GAN of [4].

5 Conclusion

We have presented a model for document image binarization that encompasses two key components: i. a Generative Adversarial architecture that is comprised of two discriminators, aimed to capture data interdependencies on a coarse as well as a finer scale of the input document image; ii. use of quaternionic layers, that replace real-valued fully connected, convolution and deconvolution layers. The end-result is a model that attains state-of-the-art performance in a multitude of binarization metrics, all the while being several times (4×) more compact than its real-valued counterpart (Fig. 1). For future work, we envisage exploring uses of more recent developments in hypercomplex architectures for binarization [35], or ways to fusion quaternion networks with other methodological paradigms, like probabilistic approaches on inference [26,27]. Also, more extensive tests are to be conducted, including other DIBCO datasets, or more recently published datasets [14] and other binarization methods [29].

Acknowledgments. This research has been partially co - financed by the EU and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the calls : “RESEARCH - CREATE - INNOVATE”, project *Culdile* (code T1EΔK - 03785) and “OPEN INNOVATION IN CULTURE”, project *Bessarion* (T6YBΠ - 00214).

References

1. Alexiadis, D.S., Daras, P.: Quaternionic signal processing techniques for automatic evaluation of dance performances from mocap data. IEEE Trans. Multimedia **16**(5), 1391–1406 (2014)
2. Ayyalasomayajula, K.R., Malmberg, F., Brun, A.: PDNet: semantic segmentation integrated with a primal-dual network for document binarization. Pattern Recogn. Lett. **121**, 52–60 (2019)
3. Chakraborty, A.: Implementation of binarization with dual discriminator GAN (2023). https://github.com/anuran-Chakraborty/BinarizationDualDiscriminatorGAN. Accessed Jan 2023
4. De, R., Chakraborty, A., Sarkar, R.: Document image binarization using dual discriminator generative adversarial networks. IEEE Signal Process. Lett. **27**, 1090–1094 (2020)
5. Dimitrakopoulos, P., Sfikas, G., Nikou, C.: Variational feature pyramid networks. In: International Conference on Machine Learning, pp. 5142–5152. PMLR (2022)
6. Ell, T.A., Sangwine, S.J.: Hypercomplex Fourier transforms of color images. IEEE Trans. Image Process. **16**(1), 22–35 (2007)
7. Fraleigh, J.B.: A First Course in Abstract Algebra, 7th (2002)

8. Gatos, B., Pratikakis, I., Perantonis, S.J.: Adaptive degraded document image binarization. Pattern Recogn. **39**(3), 317–327 (2006)
9. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall Inc., Upper Saddle River (1989)
10. Kuipers, J.B.: Quaternions and Rotation Sequences: A Primer with Application to Orbits, Aerospace and Virtual Reality. Princeton University Press, Princeton (1999)
11. Likforman-Sulem, L., Darbon, J., Smith, E.H.B.: Enhancement of historical printed document images by combining total variation regularization and non-local means filtering. Image Vis. Comput. **29**(5), 351–363 (2011)
12. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017)
13. Lins, R.D., Bernardino, R.B., Barboza, R., Oliveira, R.: The winner takes it all: choosing the ”best” binarization algorithm for photographed documents. In: Uchida, S., Barney, E., Eglin, V. (eds.) Document Analysis Systems. DAS 2022. Lecture Notes in Computer Science, vol. 13237, pp. 48–64. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06555-2_4
14. Lins, R.D., Bernardino, R.B., Smith, E.B., Kavallieratou, E.: ICDAR 2021 competition on time-quality document image binarization. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) ICDAR 2021. LNCS, vol. 12824, pp. 708–722. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-86337-1_47
15. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through l_0 regularization. arXiv preprint: arXiv:1712.01312 (2017)
16. Macías-Virgós, E., Pereira-Sáez, M., Tarrío-Tobar, A.D.: Rayleigh quotient and left eigenvalues of quaternionic matrices. Linear Multilinear Algebra, 1–17 (2022)
17. Mondal, R., Chakraborty, D., Chanda, B.: Learning 2D morphological network for old document image binarization. In: 2019 International Conference on Document Analysis and Recognition (ICDAR), pp. 65–70. IEEE (2019)
18. Nitta, T.: A quaternary version of the backpropagation algorithm. In: Proceedings of ICNN’95 - International Conference on Neural Networks, pp. 2753–2756 (1995)
19. Otsu, N.: A threshold selection method from gray-level histograms. IEEE Trans. Syst. Man Cybern. **9**(1), 62–66 (1979)
20. Papamarkos, N., Gatos, B.: A new approach for multilevel threshold selection. CVGIP: Graph. Models Image Process. **56**(5), 357–370 (1994)
21. Parcollet, T., Morchid, M., Linarès, G.: A survey of quaternion neural networks. Artif. Intell. Rev. **53**(4), 2957–2982 (2020)
22. Pratikakis, I., Zagoris, K., Barlas, G., Gatos, B.: ICDAR2017 competition on document image binarization (DIBCO 2017). In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 1395–1403. IEEE (2017)
23. Prince, S.J.: Understanding Deep Learning. MIT Press, Cambridge (2023). https://udlbook.github.io/udlbook/
24. Retsinas, G., Elafrou, A., Goumas, G., Maragos, P.: Online weight pruning via adaptive sparsity loss. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 3517–3521. IEEE (2021)
25. Sauvola, J., Pietikäinen, M.: Adaptive document image binarization. Pattern Recogn. **33**(2), 225–236 (2000)
26. Sfikas, G., Nikou, C., Galatsanos, N., Heinrich, C.: MR brain tissue classification using an edge-preserving spatially variant Bayesian mixture model. In: Metaxas,

D., Axel, L., Fichtinger, G., Székely, G. (eds.) MICCAI 2008. LNCS, vol. 5241, pp. 43–50. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-85988-8_6

27. Sfikas, G., Nikou, C., Galatsanos, N., Heinrich, C.: Majorization-minimization mixture model determination in image segmentation. In: CVPR 2011, pp. 2169–2176. IEEE (2011)
28. Sfikas, G., Retsinas, G., Giotis, A.P., Gatos, B., Nikou, C.: Keyword spotting with quaternionic ResNet: application to spotting in Greek manuscripts. In: Uchida, S., Barney, E., Eglin, V. (eds.) Document Analysis Systems. DAS 2022. Lecture Notes in Computer Science, vol. 13237, pp. 382–396. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-06555-2_26
29. Souibgui, M.A., Biswas, S., Jemni, S.K., Kessentini, Y., Fornés, A., Lladós, J., Pal, U.: DocEnTr: an end-to-end document image enhancement Transformer. In: 2022 26th International Conference on Pattern Recognition (ICPR), pp. 1699–1705. IEEE (2022)
30. Subakan, Ö.N., Vemuri, B.C.: A quaternion framework for color image smoothing and segmentation. Int. J. Comput. Vision **91**(3), 233–250 (2011)
31. Tensmeyer, C., Martinez, T.: Document image binarization with fully convolutional neural networks. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), vol. 1, pp. 99–104. IEEE (2017)
32. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
33. Vo, Q.N., Kim, S.H., Yang, H.J., Lee, G.: Binarization of degraded document images based on hierarchical deep supervised network. Pattern Recogn. **74**, 568–586 (2018)
34. Westphal, F., Lavesson, N., Grahn, H.: Document image binarization using recurrent neural networks. In: 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pp. 263–268. IEEE (2018)
35. Zhang, A., et al.: Beyond fully-connected layers with quaternions: parameterization of hypercomplex multiplications with $1/n$ parameters. In: International Conference on Learning Representations (ICLR 2021) (2021). arXiv:2102.08597
36. Zhang, F.: Quaternions and matrices of quaternions. Linear Algebra Appl. **251**, 21–57 (1997)