An Introduction to Deep Morphological Networks

KEILLER NOGUEIRA [1,2]**, JOCELYN CHANUSSOT** [3]**, (Fellow, IEEE),**
MAURO DALLA MURA [4,5]**, (Senior Member, IEEE),**
AND JEFERSSON A. DOS SANTOS [1]**, (Member, IEEE)**

[1]Department of Computer Science, Universidade Federal de Minas Gerais, Belo Horizonte 31270-901, Brazil
[2]Computing Science and Mathematics, University of Stirling, Stirling FK9 4LA, U.K.
[3]Inria, CNRS, Grenoble INP, LJK, University Grenoble Alpes, 38000 Grenoble, France
[4]CNRS, Grenoble INP, GIPSA-Laboratory, University of Grenoble Alpes, 38000 Grenoble, France
[5]Tokyo Tech World Research Hub Initiative (WRHI), School of Computing, Tokyo Institute of Technology, Tokyo 152-8550, Japan

Corresponding author: Keiller Nogueira (keiller.nogueira@stir.ac.uk)

ABSTRACT Over the past decade, Convolutional Networks (ConvNets) have renewed the perspectives of the research and industrial communities. Although this deep learning technique may be composed of multiple layers, its core operation is the convolution, an important linear filtering process. Easy and fast to implement, convolutions actually play a major role, not only in ConvNets, but in digital image processing and analysis as a whole, being effective for several tasks. However, aside from convolutions, researchers also proposed and developed non-linear filters, such as operators provided by mathematical morphology. Even though these are not so computationally efficient as the linear filters, in general, they are able to capture different patterns and tackle distinct problems when compared to the convolutions. In this paper, we propose a new paradigm for deep networks where convolutions are replaced by non-linear morphological filters. Aside from performing the operation, the proposed Deep Morphological Network (DeepMorphNet) is also able to learn the morphological filters (and consequently the features) based on the input data. While this process raises challenging issues regarding training and actual implementation, the proposed DeepMorphNet proves to be able to extract features and solve problems that traditional architectures with standard convolution filters cannot.

INDEX TERMS Convolutional networks, deep learning, deep morphological networks, mathematical morphology.
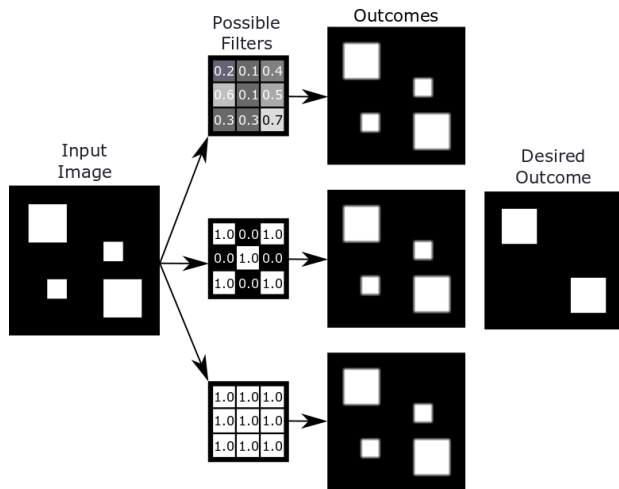
I. INTRODUCTION

Over the past decade, Convolutional Networks (ConvNet) [1] have been a game changer in the computer vision community, achieving state-of-the-art in several computer-vision applications, including image classification [2], [3], object and scene recognition [4]–[8], and many others. Although this deep learning technique may be composed of several distinct components (such as convolutional and pooling layers, non-linear activation functions, etc), its core operation is the convolution, a linear filtering process whose weights, in this case, are to be learned based on the input data. Easy and fast to implement, convolutions actually play a major role, not only in ConvNets [1], but in digital image processing and analysis [9], [10] as a whole, being effective for many tasks (including image denoising [11], edge detection [12],

etc) and employed by several techniques (such as the filtering approaches [10]).

Aside from convolutions, researchers also proposed and developed non-linear filters, such as operators provided by mathematical morphology. Even though these are not so computationally efficient as the linear filters, in general, they are able to capture different patterns and tackle distinct problems when compared to the convolutions. For instance, suppose one desires to preserve only the large objects of an input image with 4×4 and 2×2 squares. As presented in Figure 1, despite having a myriad of possible configurations, the convolution is not able to produce such an outcome that can be easily obtained, for example, by a non-linear morphological opening. In fact, supported by this capacity of extracting distinct features, some non-linear filters, such as the morphological operations [13], are still very popular and state-of-the-art in some scenarios [14]–[17].

The associate editor coordinating the review of this manuscript and approving it for publication was Geng-Ming Jiang.

FIGURE 1. Illustration showing the lack of ability of the convolution filter to produce certain outcomes that are easily generated by non-linear operations. The goal here is to preserve only the larger squares of the input image, as presented in the desired outcome. Towards such objective, this image is processed by three distinct convolution filters producing different outputs, none of them similar to the desired outcome. However, a simple morphological opening with a 3 × 3 structuring element is capable of generating such output.

In this paper, we propose a novel paradigm for deep networks where linear convolutions are replaced by the aforementioned non-linear morphological operations. Furthermore, differently from the current literature, wherein distinct morphological filters must be evaluated in order to find the most suitable ones for each application, the proposed technique, called Deep Morphological Network (DeepMorphNet), learns the filters (and consequently the features) based on the input data. Technically, the processing of each layer of the proposed approach can be divided into three steps/operations: (i) depthwise convolution [18], employed to rearrange the input pixels according to the binary filters, (ii) depthwise pooling, used to select some pixels and generate an eroded or dilated outcome, and (iii) pointwise convolution [18], employed to combine the generated maps producing one final morphological map (per neuron). This process resembles the depthwise separable convolutions [18] but using binary filters and one more step (the second one) between the convolutions. Note that, to the best of our knowledge, this is the first proof of concept work related to networks capable of performing and optimizing exact (non-approximate) morphological operations with **flat** structuring elements (i.e., filters). Particularly, this is an advantage given that the vast majority of the mathematical morphology operators based on structuring elements employ flat filters. While this replacement process raises challenging issues regarding training and actual implementation, the proposed DeepMorphNet proves to be able to solve problems that traditional architectures with standard convolution filters cannot.

In practice, we can summarize the main contributions of this paper as follows:

- a novel paradigm for deep networks where linear convolutions are replaced by the non-linear morphological operations, and
- a technique, called Deep Morphological Network (DeepMorphNet), capable of performing and optimization morphological operations.

The remainder of this paper is organized as follows. Section II introduces some background concepts and presents the related work. The proposed method is presented in Section III. The experimental setup is introduced in Section IV while Section V presents the obtained results. Finally, Section VI concludes the paper.

II. BACKGROUND KNOWLEDGE AND RELATED WORK
This section introduces the basic principles underlying mathematical morphology, and reviews the main approaches that exploit such operations for distinct image tasks.

A. MATHEMATICAL MORPHOLOGY
Morphological operations, commonly employed in the image processing area, are strongly based on mathematical morphology. Since its introduction to the image domain, these morphological operations have been generalized from the analysis of a single band image to hyperspectral images made up of hundreds of spectral channels and has become one of the state-of-the-art techniques for a wide range of applications [13]. This study area includes several different operations (such as erosion, dilation, opening, closing, top-hats, and reconstruction), which can be applied to binary and grayscale images in any number of dimensions [13].

Formally, consider a grayscale 2D image $I(\cdot)$ as a mapping from the coordinates ($\mathbb{Z}^2$) to the pixel-value domain ($\mathbb{Z}$). Most morphological transformations process this input image I using a structuring element (SE) (usually defined prior to the operation). A flat[1] SE $B(\cdot)$ can be defined as a function that, based on its format (or shape), returns a set of neighbors of a pixel (i, j). This neighborhood of pixels is taken into account during the morphological operation, i.e., while probing the image I. As introduced, the definition of the SE is of vital importance for the process to extract relevant features. However, in literature [19], [20], this definition is performed experimentally (with common shapes being squares, disks, diamonds, crosses, and x-shapes), an expensive process that does not guarantee a good descriptive representation.

After its definition, the SE can be then employed in several morphological processes. Most of these operations are usually supported by two basic morphological transformations: *erosion* $\mathcal{E}(\cdot)$ and *dilation* $\delta(\cdot)$. Such operations receive basically the same input: an image I and the SE B. While erosion transformations process each pixel (i, j) using the supremum function $\wedge$, as denoted in Equation 1, the dilation operations process the pixels using the infimum $\vee$ function, as presented in Equation 2. Intuitively, these two operations

---
[1]A flat SE is binary and only defines which pixels of the neighborhood should be taken into account. On the other hand, a non-flat SE contains finite values used as additive offsets in the morphological computation.

probe an input image using the SE, i.e., they position the structuring element at all possible locations in the image and analyze the neighborhood pixels. This process, somehow similar to the convolution procedure, outputs another image with regions compressed or expanded. Some examples of erosion and dilation are presented in Figure 2, in which it is possible to notice the behavior of each operation. As can be noticed erosion affects brighter structures while dilation influences darker ones (w.r.t. the neighborhood defined by the SE).

$$\mathcal{E}(B, I)_{(i,j)} = \{\wedge I((i,j)')|(i,j)' \in B(i,j) \cup I(i,j)\} \quad (1)$$

$$\delta(B, I)_{(i,j)} = \{\vee I((i,j)')|(i,j)' \in B(i,j) \cup I(i,j)\} \quad (2)$$

If we have an ordered set, then the erosion and dilation operations can be simplified. This is because the infimum and the supremum are respectively equivalent to the minimum and the maximum functions when dealing with ordered sets. In this case, erosion and dilation can be defined as presented in Equations 3 and 4, respectively.

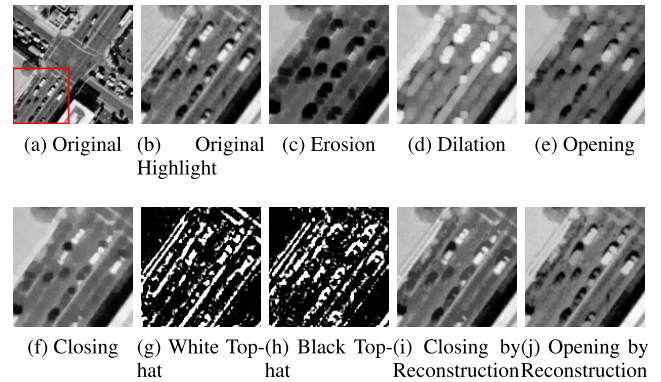$$\mathcal{E}(B, I)_{(i,j)} = \{\min_{(i,j)' \in B(i,j)} (I((i,j)'))\} \quad (3)$$

$$\delta(B, I)_{(i,j)} = \{\max_{(i,j)' \in B(i,j)} (I((i,j)'))\} \quad (4)$$

Based on these two fundamental transformations, other more complex morphological operations may be computed. The morphological *opening*, denoted as $\gamma(\cdot)$ and defined in Equation 5, is simply an erosion operation followed by the dilation (using the same SE). In contrast, a morphological *closing* $\varphi(\cdot)$ of an image, defined in Equation 6, is a dilation followed by the erosion (using the same SE). Intuitively, an opening flattens bright objects that are smaller than the size of the SE and, because of dilation, mostly preserves the bright large areas. A similar conclusion can be drawn for darker structures when closing is performed. Examples of this behavior can be seen in Figure 2. It is important to highlight that by using erosion and dilation transformations, opening and closing perform geodesic reconstruction in the image. Operations based on this paradigm belongs to the class of filters that operate only on connected components (flat regions) and cannot introduce any new edge to the image. Furthermore, if a segment (or component) in the image is larger than the SE then it will be unaffected, otherwise, it will be merged to a brighter or darker adjacent region depending upon whether a closing or opening is applied. This process is crucial because it avoids the generation of distorted structures, which is obviously an undesirable effect.

$$\gamma(B, I) = \delta(B, \mathcal{E}(B, I)) \quad (5)$$

$$\varphi(B, I) = \mathcal{E}(B, \delta(B, I)) \quad (6)$$

Other important morphological operations are the *top-hats*. Top-hat transform is an operation that extracts small elements and details from given images. There are two types of top-hat transformations: (i) the white one $\mathcal{T}^w(\cdot)$, defined in Equation 7, in which the difference between the input image and its

(a) Original (b) Original Highlight (c) Erosion (d) Dilation (e) Opening

(f) Closing (g) White Top-hat (h) Black Top-hat (i) Closing by Reconstruction (j) Opening by Reconstruction

FIGURE 2. Examples of morphological images generated for the UCMerced Land-use Dataset. For better viewing and understanding, images (b)-(j) only present a left-bottom zoom of the original image (a). All these images were processed using a 5 × 5 square as structuring element.

opening is calculated, and (ii) the black one, denoted as $\mathcal{T}^b(\cdot)$ and defined in Equation 8, in which the difference between the closing and the input image is performed. White top-hat operation preserves elements of the input image brighter than their surroundings but smaller than the SE. On the other hand, black top-hat maintains objects smaller than the SE with brighter surroundings. Examples of these two operations can be seen in Figure 2.

$$\mathcal{T}^w(B, I) = I - \gamma(B, I) \quad (7)$$

$$\mathcal{T}^b(B, I) = \varphi(B, I) - I \quad (8)$$

Another important morphological operation based on erosions and dilations is the *geodesic reconstruction*. There are two types of geodesic reconstruction: by erosion and by dilation. The geodesic reconstruction by erosion $\rho^{\mathcal{E}}(\cdot)$, mathematically defined in Equation 9, receives two parameters as input: an image I and a SE B. The image I (also referenced in this operation as mask image) is dilated by the SE B ($\delta(B, I)$) creating the marker image Y ($Y \in I$), responsible for delimiting which objects will be reconstructed during the process. A SE B' (usually with any elementary composition [13]) and the marker image Y are provided for the reconstruction operation $\mathcal{R}_I^{\mathcal{E}}(\cdot)$. This transformation, defined in Equation 10, reconstructs the marker image Y (with respect to the mask image I) by recursively employing geodesic erosion (with the elementary SE B') until idempotence is reached (i.e., $\mathcal{E}_I^{(n)}(\cdot) = \mathcal{E}_I^{(n+1)}(\cdot)$). In this case, a geodesic erosion $\mathcal{E}_I^{(1)}(\cdot)$, defined in Equation 11, consists of a pixel-wise maximum operation between an eroded (with elementary SE B') marker image Y and the mask image I. By duality, a geodesic reconstruction by dilation can be defined, as presented in Equation 12. These two crucial operations try to preserve all large (than the SE) objects of the image removing bright and dark small areas, such as noises. Some examples of these operations can be seen in Figure 2.

$$\rho^{\mathcal{E}}(B, I) = \mathcal{R}_I^{\mathcal{E}}(B', Y) = \mathcal{R}_I^{\mathcal{E}}(B', \delta(B, I)) \quad (9)$$

$$\mathcal{R}_I^{\mathcal{E}}(B', Y) = \mathcal{E}_I^{(n)}(B', Y)$$

$$= \underbrace{\mathcal{E}_I^{(1)}\Big(B', \mathcal{E}_I^{(1)}\big(B', \cdots \mathcal{E}_I^{(1)}\big(B', \mathcal{E}_I^{(1)}(B', Y)\big)\big)\Big)}_{n \text{ times}}$$

(10)

$$\mathcal{E}_I^{(1)}(B', Y) = \max\{\mathcal{E}(B', Y), I\} \tag{11}$$

$$\rho^{\delta}(B, I) = \mathcal{R}_I^{\delta}(B', Y) = \mathcal{R}_I^{\delta}(B', \mathcal{E}(B, I)) \tag{12}$$

Note that geodesic reconstruction operations require an iterative process until the convergence. This procedure can be expensive, mainly when working with a large number of images. An approximation of such operations, presented in Equations 13 and 14, can be achieved by performing just **one** transformation over the marker image with a large (than the SE used to create the marker image) structuring element. In other words, suppose that B is the SE used to create the marker image, then B', the SE used in the reconstruction step, should be larger than B. This process is faster since only one iteration is required, but may lead to worse results, given that the use of a large filter can make the reconstruction join objects that are close in the scene (a phenomenon known as *leaking* [13]).

$$\tilde{\rho}^{\mathcal{E}}(B, I) = \mathcal{E}_I(B', \delta(B, I)) \tag{13}$$

$$\tilde{\rho}^{\delta}(B, I) = \delta_I(B', \mathcal{E}(B, I)) \tag{14}$$

B. RELATED WORK

As introduced, such non-linear morphological operations [13] have the ability to preserve some features that may be essential for some problems. Supported by this, several tasks and applications have exploited the benefits of morphological operations, such as image analysis [16], [17], [21]–[24], classification [25]–[27], segmentation [14], [15], [20], [28], [29], and so on.

Some of these techniques [16], [21], [22], [29] are strongly based on mathematical morphology. These approaches process the input images using only morphological operations. The produced outcomes are then analyzed in order to extract high-level semantic information, such as borders, area, geometry, volume, and more. Other works [14], [15], [17], [19], [27] go further and use morphology to extract robust features that are employed as input to machine learning techniques (such as Support Vector Machines, and decision trees) to perform image analysis, classification, and segmentation. Usually, in these cases, the input images are processed using several different morphological transformations, each one employing a distinct structuring element, in order to improve the diversity of the extracted features. All these features are then concatenated and used as input for the machine learning techniques.

More recently, ConvNets [1] started achieving outstanding results, mainly in applications related to images. Therefore, it would be more than natural for researchers to propose works combining the benefits of ConvNets and

morphological operations. In fact, several works [20], [25], [26], [28] tried to combine these techniques to create a more robust model. Some works [20], [28] employed morphological operations as a pre-processing step in order to extract the first set of discriminative features. In these cases, pre-defined (hand-crafted) structuring elements are employed. Those techniques use such features as input for a ConvNet responsible to perform the classification. Based on the fact that morphology generates interesting features that are not captured by the convolutional networks, such works achieved outstanding results on pixelwise classification.

Other works [23]–[26] introduced morphological operations into neural networks, creating a framework in which the structuring elements are optimized. Masci *et al.* [25] proposed a convolutional network that aggregates pseudo-morphological operations. Specifically, their proposed approach uses the counter-harmonic mean, which allows the convolutional layer to perform its traditional linear process, or approximations of morphological operations. They show that the approach produces outcomes very similar to real morphological operations. Mellouli *et al.* [26] performed a more extensive validation of the previous method, proposing different deeper networks that are used to perform image classification. In their experiments, the proposed network achieved promising results for two datasets of digit recognition. In [23], the authors proposed a new network capable of performing some morphological operations (including erosion, dilation, opening, and closing) while optimizing non-flat structuring elements. Their proposed network, evaluated for image de-raining and de-hazing, produced results similar to those of a ConvNet but using much fewer parameters. Finally, Franchi *et al.* [24] proposed a new deep learning framework capable of performing non-approximated mathematical morphology operations (including erosion, dilation) while optimizing non-flat structuring elements. Their method produced competitive results for edge detection and image denoising when compared to classical techniques and standard ConvNets.

In this work, we proposed a new network capable of performing and optimizing several morphological operations, including erosion, dilation, openings, closing, top-hats, and an approximation of geodesic reconstructions. Several differences may be pointed out between the proposed approach and the aforementioned works: (i) differently from [25], [26], the proposed technique really carries out morphological operations without any approximation (except for the reconstruction), (ii) the morphological operations incorporated into the proposed network use flat SEs (which may be used with any input image), thus differing from other methods, such as [23], [24], that can only optimize morphological operations using non-flat SEs and, consequently, can only be used with grayscale input data, and (iii) to the best of our knowledge, this is the first approach to implement (approximate) morphological geodesic reconstruction within deep-learning based models.

III. DEEP MORPHOLOGICAL NETWORKS

In this section, we present the proposed approach, called Deep Morphological Networks (or simply DeepMorphNets), that replaces the linear convolutional filters with optimizable non-linear morphological operations. Such replacement allows the proposed network to capture distinct information when compared to previous existing models, an advantage that may assist different applications. However, this process raises several challenging issues.

One first challenge is due to the linear nature of the core operations of the existing networks. The convolutional layers extract features from the input data using an optimizable filter by performing only linear operations not supporting non-linear ones. Formally, let us assume a 3D input $y(\cdot)$ of a convolutional layer as a mapping from coordinates ($\mathbb{Z}^3$) to the pixel-value domain ($\mathbb{Z}$ or $\mathbb{R}$). Analogously, the trainable filter (or weight) $W(\cdot)$ of such layer can be seen as a mapping from 3D coordinates ($\mathbb{Z}^3$) to the real-valued weights ($\mathbb{R}$). A standard convolutional layer performs a convolution of the filter $W(\cdot)$ over the input $y(\cdot)$, according to Equation 15. Note that the output of this operation is the summation of the linear combination between input and filter (across both space and depth). Also, observe the difference between this operation and the morphological ones stated in Section II-A. This shows that replacing the convolutional filters with morphological operations is not straightforward.

$$S(W, y)_{(i,j)} = \sum_m \sum_n \sum_l W(m, n, l) y(i+m, j+n, l) \quad (15)$$

Another important challenge is due to the optimization of non-linear operations by the network. Precisely, in ConvNets, a loss function $\mathcal{L}$ is used to optimize the model. Nevertheless, the objective of any network is to minimize this loss function by adjusting the trainable parameters (or filters) W. Such optimization is traditionally based on the derivatives of the loss function $\mathcal{L}$ with respect to the weights W. For instance, suppose Stochastic Gradient Descent (SGD) [1] is used to optimize a ConvNet. As presented in Equation 16, the optimization of the filters depends directly on the partial derivatives of the loss function $\mathcal{L}$ with respect to the weights W (employed with a pre-defined learning rate α). Those partial derivatives are usually obtained using the backpropagation algorithm [1], which is strongly supported by the fact that all operations of the network are easily differentiable, including the convolution presented in Equation 15. However, this algorithm cannot be directly applied to non-linear operations, such as the presented morphological ones, because those operations do not have easy derivatives.

$$W = W - \alpha \frac{\partial \mathcal{L}}{\partial W} \quad (16)$$

Overcoming such challenges, we propose a network that employs depthwise and pointwise convolution with depthwise pooling to recreate and optimize morphological operations. The basic concepts of the proposed technique

is presented in Section III-A. This concept is employed to create morphological neurons and layers, presented in Sections III-B and III-C, respectively. Section III-D explains the optimization processed performed to learn the structure elements. Finally, the morphological architecture exploited in this work are introduced in Section III-E.

A. BASIC MORPHOLOGICAL FRAMEWORK

Towards the preservation of the end-to-end learning strategy, we propose a new framework, capable of performing morphological erosion and dilation, that uses operations already employed in other existing deep learning-based methods. The processing of this framework can be separated into two steps. The **first** one employs depthwise convolution [18] to perform a delimitation of features, based on the neighborhood (or filter). As defined in Equation 17, this type of convolution differs from standard ones since it handles the input depth independently, using the same filter W to every input channel. In other words, suppose that a layer performing depthwise convolution has k filters and receives an input with l channels, then the processed outcome would be an image of $k \times l$ channels, since each k-th filter would be applied to each l-th input channel. The use of depthwise convolution simplifies the introduction of morphological operations into the deep network since the linear combination performed by this convolution does not consider the depth (as in standard convolutions presented in Equation 15). This process is fundamental for the recreation of morphological operations since such transformations can only process one single channel at a time.

$$S_l(W, y)_{(i,j)} = \sum_m \sum_n W(m, n) y(i+m, j+n, l) \quad (17)$$

However, just using this type of convolution does not allow the reproduction of morphological transformations, given that a spatial linear combination is still performed by this convolutional operation. To overcome this, all filters W are first converted into binary and then used in the depthwise convolution operation. This binarization process, referenced hereafter as max-binarize, activates only the highest value of the filter. Formally, the max-binarize $b(\cdot)$ is a function that receives as input the real-valued weights W and processes them according to Equation 18, where $\mathbb{1}\{condition\}$ is the indicator function. This process outputs a binary version of the weights, W^b, in which **only** the highest value in W is activated in W^b. By using W^b, the linear combination performed by depthwise convolution can be seen as a simple operation that preserves the exact value of the single pixel activated by this binary filter.

$$W_{(i,j)}^b = b(W(i,j)) = \mathbb{1}\{max_{m,n}(W(m, n)) = W(i, j)\} \quad (18)$$

But preserving only one pixel with respect to the binary filter is not enough to reproduce the morphological operations, since they usually operate over a neighborhood (defined by the SE B). In order to reproduce this neighborhood concept in the depthwise convolution operation, we decompose each
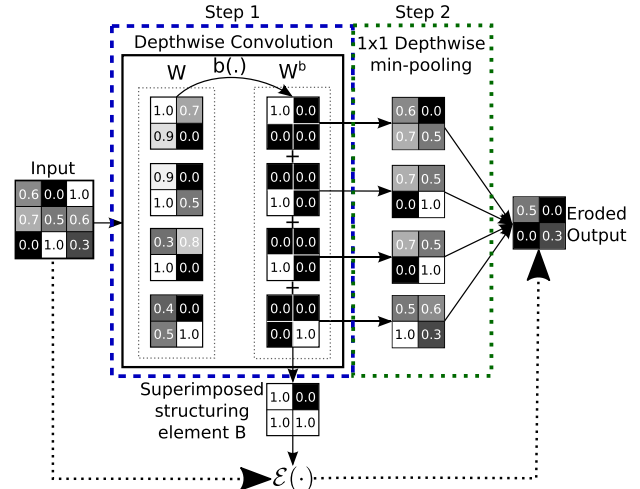
filter W into several ones, that when superimposed retrieve the final SE B. More specifically, suppose a filter W with size $s \times s$. Since only one position can be activated at a time, this filter has a total of s^2 possible activation variations. This analysis is the same if we consider each element of a $s \times s$ SE independently. Based on all this, a set of s^2 max-binary filters with size $s \times s$ is able to cover all possible configurations of a SE with the same size. Thus, a set of s^2 filters with size $s \times s$ can be seen as a decomposed representation of the structuring element, given that those s^2 filters (with only a single activated position) can be superimposed in order to retrieve any possible $s \times s$ neighborhood configuration defined by the SE. By doing this, the concept of neighborhood introduced by the SE can be exploited by the depthwise convolution. Technically, a s^2 set of $s \times s$ filters W can be converted into binary weights W^b and then, used to process the input data. When exploited by Equation 17, each of these s^2 binary filter W^b will preserve only one pixel which is directly related to one specific position of the neighborhood. As may be observed, this first step recreates, in depth, the neighborhood of a pixel delimited by a $s \times s$ SE B, which is essentially represented by s^2 binary filters W^b of size $s \times s$.

Since the SE B was decomposed in depth, in order to retrieve it, a depthwise operation must be performed over the s^2 binary filters W^b. Analogously, a depthwise operation is also required to retrieve the final outcome, i.e., the eroded or dilated image. This is the **second** step of this proposed framework, which is responsible to extract the relevant information based on the depthwise neighborhood. In this step, an operation, called depthwise pooling $P(\cdot)$, processes the s^2 outcomes (of the decomposed filters), producing the final morphological outcome. This pooling operation is able to actually output the morphological erosion and dilation by using pixel and depthwise minimum and maximum functions, as presented in Equations 19 and 20, respectively. The outcome of this second step is the final (eroded or dilated) feature map that will be exploited by any subsequent process.

$$P^{\mathcal{E}}(y)_{(i,j)} = \min_l y(i,j,l) \qquad (19)$$

$$P^{\delta}(y)_{(i,j)} = \max_l y(i,j,l) \qquad (20)$$

Equations 21 and 22 compile the two steps performed by the proposed framework for morphological erosion and dilation, respectively. This operation, denoted here as $M(\cdot)$, performs a depthwise convolution (first step), which uses max-binary filters that decompose the representation of the neighborhood concept introduced by SEs, followed by a pixel and depthwise pooling operation (second step), outputting the final morphological (eroded or dilated) feature maps. Note the similarity between these functions and Equations 3 and 4 presented in Section II-A. The main difference between these equations is in the neighborhood definition. While in the standard morphology, the neighborhood of a pixel is defined spatially (via SE B), in the proposed framework, the neighborhood is defined along the channels due to the decomposition of the SE B into several filters and, therefore, minimum and

FIGURE 3. Example of a morphological erosion based on the proposed framework. The 4 filters W (with size 4 × 4) actually represent a unique 4 × 4 SE. Each filter W is first converted to binary W^b, and then used to process each input channel (step 1, blue dashed rectangle). The output is then processed via a pixel and depthwise min-pooling to produce the final eroded output (step 2, green dotted rectangle). Note that the binary filters W^b, when superimposed, retrieve the final SE B. The dotted line shows that the processing of the input with the superimposed SE B using the standard morphological erosion results in the same eroded output image produced by the proposed morphological erosion.

maximum operations also operate over the channels.

$$M^{\mathcal{E}}(W,y)_{(i,j)} = P^{\mathcal{E}}(S_l(W,y))_{(i,j)} = \min_l S_l(W,y)_{(i,j)} \qquad (21)$$

$$M^{\delta}(W,y)_{(i,j)} = P^{\delta}(S_l(W,y))_{(i,j)} = \max_l S_l(W,y)_{(i,j)} \qquad (22)$$

A visual example of the proposed framework being used for morphological erosion is presented in Figure 3. In this example, the depthwise convolution has 4 filters W with size 4×4 which actually represent a unique 4×4 SE. The filters W are first converted into binary using the max-binarize function $b(\cdot)$, presented in Equation 18. Then, each binary filter W^b is used to process (step 1, blue dashed rectangle) each input channel (which, for simplicity, is only one in this example) using Equation 17. In this process, each binary filter W^b outputs an image in which each pixel has a direct connection to the one position activated in that filter. The output is then processed (step 2, green dotted rectangle) via a pixel and depthwise min-pooling $P(\cdot)^{\mathcal{E}}$ (according to Equation 19) to produce the final eroded output. Note that the binary filters W^b, when superimposed retrieve the final SE B. The dotted line shows that the processing of the input with the superimposed SE B using the standard erosion ($\mathcal{E}(\cdot)$ presented in Equation 3) results in the same eroded output image produced by the proposed morphological erosion.

B. MORPHOLOGICAL PROCESSING UNITS

The presented framework is the foundation of all proposed morphological processing units (or neurons). Before presenting them, it is important to observe that, although the proposed framework is able to reproduce morphological erosion and dilation, it has an important drawback: since it

employs depthwise convolution, the number of outcomes can grow potentially, given that, each input channel is processed independently by each processing unit. In order to overcome this issue and make the proposed technique more scalable, we propose to use a pointwise convolution [18] to force each processing unit to output only one image (or feature map). Particularly, all neurons proposed in this work have the same design with two parts: (i) the core operation (fundamentally based on the proposed framework), in which the processing unit performs its morphological transformation outputting multiple outcomes, and (ii) the pointwise convolution [18], which performs a pixel and depthwise (linear) combination of the outputs producing only one outcome. Observe that, even though the pointwise convolution performs a depthwise combination of the multiple outcomes, it does not learn any spatial feature, since it employs a pixelwise (or pointwise) operation, managing each pixel separately. This design allows the morphological neuron to have the exact same input and output of a standard existing processing unit, i.e., it receives as input an image with any number of bands and outputs a single new representation. It is interesting to notice that this processing unit design employs depthwise and pointwise convolution [18], resembling very much the depthwise separable convolutions [18], but with extra steps and binary decomposed filters. Next Sections explain the **core operation** of all proposed morphological processing units. Note that, although not mentioned in the next Sections, the pointwise convolution is present in all processing units as aforementioned.

1) COMPOSED PROCESSING UNITS
The first morphological neurons, called composed processing units, have, in their core, a morphological erosion followed by a dilation (or vice-versa), without any constraint on the weights. The motivation behind the composed processing unit is based on the potential of the learned representation. While erosion and dilation can learn simple representations, the combination of these operations is able to capture more complex information. Equations 23 and 24 present the two possible configurations of the morphological composed neurons. It is important to notice that the weights (W_1 and W_2) of each operation of this neuron are independent.

$$M^{C_\delta}(W, y) = M^\delta(W_2, M^\mathcal{E}(W_1, y)) \qquad (23)$$
$$M^{C_\mathcal{E}}(W, y) = M^\mathcal{E}(W_2, M^\delta(W_1, y)) \qquad (24)$$

2) OPENING AND CLOSING PROCESSING UNITS
The proposed framework is also able to support the implementation of more complex morphological operations. The most intuitive and simple transformations to be implemented are the opening and closing (presented in Section II-A). In fact, the implementation of the opening and closing processing units, using the proposed framework, is straightforward. The core of such neurons is very similar to that of the composed processing units, except that in this case a tie on the filters of the two basic morphological operations is required in order to make them use the same weights,

i.e., the same SE *B*. Equations 25 and 26 define the opening and closing morphological neurons, respectively. Note the similarity between these functions and Equations 5 and 6.

$$M^\gamma(W, y) = M^\delta(W, M^\mathcal{E}(W, y)) \qquad (25)$$
$$M^\varphi(W, y) = M^\mathcal{E}(W, M^\delta(W, y)) \qquad (26)$$

3) TOP-HAT PROCESSING UNITS
The implementation of other, more complex, morphological operations is a little more tricky. This is the case of the top-hat operations, which require both the input and processed data to generate the final outcome. Therefore, for such operations, a skip connection [1] (based on the identity mapping) is employed to support the forwarding of the input data, allowing it to be further processed. The core of the top-hat processing units is composed of three parts: (i) an opening or closing morphological processing unit, depending on the type of the top-hat, (ii) a skip connection, that allows the forwarding of the input data, and (iii) a subtraction function that operates over the data of both previous parts, generating the final outcome. Such operation and its counterpart (the black top-hat) are defined in Equations 27 and 28, respectively.

$$M^{\mathcal{T}^w}(W, y) = y - M^\gamma(W, y) \qquad (27)$$
$$M^{\mathcal{T}^b}(W, y) = M^\varphi(W, y) - y \qquad (28)$$

4) GEODESIC RECONSTRUCTION PROCESSING UNITS
Similarly to the previous processing units, the geodesic reconstruction also requires the input and processed data in order to produce the final outcome. Hence, the implementation of this important operation is also based on skip connections. Aside from this, as presented in Section II-A, reconstruction operations require an iterative process. Although this procedure is capable of producing better outcomes, its introduction in a deep network is not straightforward, given that each process can take a different number of iterations. Supported by this, the reconstruction processing units proposed in this work are an approximation, in which just one transformation over the marker image is performed. Based on this, the input is processed by two basic morphological operations (without any iteration) and an elementwise max- or min-operation (depending on the reconstruction) is performed over the input and processed images. Such concept is presented in Equations 29 and 30 for reconstruction by erosion and dilation, respectively. Note that the SE used in the reconstruction of the marker image (denoted in Section II-A by B') is a dilated version of the SE employed to create such image.

$$M^{\tilde{\rho}^\mathcal{E}}(W, y) = M_y^\mathcal{E}(W, M^\delta(W, y)) \qquad (29)$$
$$M^{\tilde{\rho}^\delta}(W, y) = M_y^\delta(W, M^\mathcal{E}(W, y)) \qquad (30)$$

C. MORPHOLOGICAL LAYER
After defining the processing units, we are able to specify the morphological layers, which provide the essential tools

for the creation of the DeepMorphNets. Similar to the standard convolutional layer, this one is composed of several processing units. However, the proposed morphological layer has two main differences when conceptually compared to the convolutional one. The first one is related to the neurons that compose the layers. Particularly, in convolutional layers, the neurons are able to perform the convolution operation. Though the filter of each neuron can be different, the operation performed by each processing unit in a convolutional layer is a simple convolution. On the other hand, there are several types of morphological processing units, from opening and closing to geodesic reconstruction. Therefore, a single morphological layer can be composed of several neurons that may be performing different operations. This process allows the layer to produce distinct (and possibly complementary) outputs, increasing the heterogeneity of the network and, consequently, the generalization capacity. The second difference is the absence of activation functions. More specifically, in modern architectures, convolutional layers are usually composed of a convolution operation followed by an activation function (such as ReLU [30]), that explicitly maps the data into a non-linear space. In morphological layers, there are only processing units and no activation function is employed.

D. OPTIMIZATION

Aside from defining the morphological layer, as introduced, we want to optimize its parameters, i.e., the filters W. *Since the proposed morphological layer uses common (differentiable) operations already employed in other existing deep learning-based methods, the optimization of the filters is straightforward.* In fact, the same traditional existing techniques employed in the training of any deep learning-based approach, such feedforward, backpropagation and SGD [1], can also be used for optimizing a network composed of morphological layers.

The training procedure is detailed in Algorithm 1. Given the training data (y_0, y^*), the **first step** is the feedforward, comprised in the loop from line 2 to 8. In the first part of line 4, the weights of the first depthwise convolution are converted into binary (according to Equation 18). Then, in the second part, the first depthwise convolution (denoted here as $*$) is performed, with the first depthwise pooling being executed in the third part of this line. The same operations are repeated in line 6 for the second depthwise convolution and pooling. Finally, in line 7, the pointwise convolution is carried out. After the forward propagation, the total error of the network can be estimated. With this error, the gradients of the last layer can be directly estimated (line 10). These gradients can be used by the backpropagation algorithm to calculate the gradients of the inner layers. In fact, this is the process performed in the **second training step**, comprised in the loop from line 11 to 15. It is important to highlight that during the backpropagation process, the gradients are calculated normally, using real-valued numbers (and not binary). Precisely, line 12 is responsible for the optimization of the

pointwise convolution. The first part (of line 16) propagates the error of a specific pointwise convolution to the previous operation, while the second part calculates the error of that specific pointwise convolution operation. The same process is repeated for the second and then for the first depthwise convolutions (lines 13 and 14, respectively). Note that during the backpropagation, the depthwise pooling is not optimized since this operation has no parameters and only passes the gradients to the previous layer. The **last step** of the training process is the update of the weights and optimization of the network. This process is comprised in the loop from line 17 to 21. For a specific layer, line 18 updates the weights of the pointwise convolution while lines 19 and 20 update the parameters of the first and second depthwise convolutions, respectively.

Algorithm 1 Training a Deep Morphological Network With L Layers

---

Require: a minibatch of inputs and targets (y_0, y^*), previous weights W, and previous learning rate α.
Ensure: updated weights W.
1: 1. Forward propagation:
2: **for** k=1 to L **do**
3:     {First Processing Unit Operation}
4:     $W_k^{b^{(1)}} \leftarrow b(W_k^{(1)})$
5:     $s_k^{(1)} \leftarrow y_{k-1} * W_k^{b^{(1)}}$
6:     $y_k^{(1)} \leftarrow P(s_k^{(1)})$
7:     {Second Processing Unit Operation}
8:     $W_k^{b^{(2)}} \leftarrow b(W_k^{(2)})$
9:     $s_k^{(2)} \leftarrow y_k^{(1)} * W_k^{b^{(2)}}$
10:     $y_k^{(2)} \leftarrow P(s_k^{(2)})$
11:     $y_k \leftarrow y_k^{(2)} * W_k^{(1\times1)}$ {Pointwise Convolution}
12: **end for**
13: 2. Backpropagation: {Gradients are not binary.}
14: Compute $g_{y_L^{(1)}} = \frac{\partial \mathcal{L}}{\partial y_L}$ knowing y_L and y^*
15: **for** k=L to 1 **do**
16:     $g_{y_{k-1}} \leftarrow g_{y_k^{(1)}} W_{k-1}^{(1\times1)}, g_{W_{k-1}^{(1\times1)}} \leftarrow g_{y_k^{(1)}}^{\mathsf{T}} y_{k-1}$
17:     $g_{y_{k-1}^{(2)}} \leftarrow g_{y_{k-1}} W_{k-1}^{b^{(2)}}, g_{W_{k-1}^{b^{(2)}}} \leftarrow g_{y_{k-1}^{(2)}}^{\mathsf{T}} y_{k-1}^{(2)}$
18:     $g_{y_{k-1}^{(1)}} \leftarrow g_{y_{k-1}^{(2)}} W_{k-1}^{b^{(1)}}, g_{W_{k-1}^{b^{(1)}}} \leftarrow g_{y_{k-1}^{(2)}}^{\mathsf{T}} y_{k-1}^{(1)}$
19: **end for**
20: 3. Update the weights:
21: **for** k=1 to L **do**
22:     $W_k^{(1\times1)} \leftarrow W_k^{(1\times1)} - \alpha g_{W_k^{(1\times1)}}$
23:     $W_k^{(1)} \leftarrow W_k^{(1)} - \alpha g_{W_k^{b^{(1)}}}$
24:     $W_k^{(2)} \leftarrow W_k^{(2)} - \alpha g_{W_k^{b^{(2)}}}$
25: **end for**

E. DeepMorphNet ARCHITECTURES

Two networks, composed essentially of morphological and fully connected layers, were proposed for the image and pixel classification tasks. Although such architectures have distinct

designs, the pointwise convolutions exploited in the morphological layers have always the same configuration: kernel 1×1, stride 1, and no padding. Furthermore, all networks receive input images with 224×224 pixels, use cross-entropy as loss function, and SGD as optimization algorithm [1]. For the pixel classification task, the proposed networks were exploited based on the pixelwise paradigm defined by [31], in which each pixel is classified independently through a context window.

The first network is the simplest one, having just a unique layer composed of one morphological opening M^{γ}. This architecture was designed to be used with the proposed synthetic datasets (presented in Section IV-A1). Because of this, it is referenced hereafter as **DeepMorphSynNet**. Note that this network was only conceived to validate the learning process of the proposed framework as explained in Section V-A.

To analyze the effectiveness of the technique in a more complex scenario, we proposed a larger network inspired by the famous AlexNet [2] architecture. It is important to highlight that AlexNet [2] was our inspiration, and not a more complex architecture, because if its simplicity, which allows a clear analysis of the benefits of the proposed technique, thus avoiding confusing them with other advantages of more complex deep learning approaches. This proposed morphological version of the AlexNet [2], called **DeepMorphNet** and presented in Figure 4, has the same number of layers of the original architecture but fewer neurons in each layer. Specifically, this network has 5 morphological and 3 fully connected layers, responsible to learn high-level features and perform the final classification. To further evaluated the potential of the proposed technique, in some experiments, a new version of the DeepMorphNet, using a modern component called Selective Kernels (SK) [32], was developed and experimented. This new network, referenced hereafter as **DeepMorphNet-SK**, uses such components to weigh the features maps, giving more attention to some maps than the others.

IV. EXPERIMENTAL SETUP

In this section, we present the experimental setup. Section IV-A presents the datasets. Baselines are described in Section IV-B while the protocol is introduced in Section IV-C.

A. DATASETS

Six datasets were employed to validate the proposed DeepMorphNets. Two image classification synthetic ones were exclusively designed to check the feature learning of the proposed technique. Other two image classification datasets were selected to further verify the potential of DeepMorphNets. Finally, to assess the performance of the proposed technique in distinct scenarios, two pixel classification datasets were exploited.

1) IMAGE CLASSIFICATION DATASETS
a: SYNTHETIC DATASETS
As introduced, two simple synthetic (image classification) datasets were designed in this work to validate the feature learning process of the proposed DeepMorphNets. In order to allow such validation, these datasets were created so that it is possible to define, a priori, the optimal SE (i.e., the SE that would produce the best results) for a classification scenario. Hence, in this case, the validation would be performed by comparing the learned SE with the optimal one, i.e., if both SEs are similar, then the proposed technique is able to perform well the feature learning step.
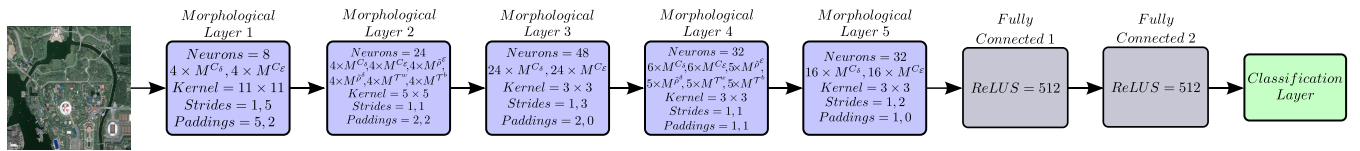
Specifically, both datasets are composed of **1,000 grayscale images** with a resolution of 224×224 pixels (a common image size employed in famous architecture such as AlexNet [2]) **equally** divided into two classes.

The **first dataset** has two classes. The first one is composed of images with small (5×5 pixels) squares whereas the second consists of images with large (9×9 pixels) squares. Each image of this dataset has only one square (of one of the above classes) positioned randomly. In this case, an opening with a SE larger than 5×5 but smaller than 9×9 should erode the small squares while keeping the others, allowing the model to perfectly classify the dataset.

More difficult, the **second synthetic dataset** has two classes of rectangles. The first class has shapes of 7×3 pixels while the other one is composed of rectangles of 3×7. As in the previous dataset, each image of this dataset has only one rectangle (of one of the above classes) positioned randomly. This case is a little more complicated because the network should learn a SE based on the orientation of one the rectangles. Particularly, it is possible to perfectly classify this dataset using a single opening operation with one of the following types of SEs: (i) a rectangle of at least 7 pixels of width and height larger than 3 but smaller than 7 pixels, which would erode the first class of rectangles and preserve the second one, or (ii) a rectangle with a width larger than 3 but smaller than 7 pixels and height larger than 7 pixels, which would erode the second class of rectangle while keeping the first one.

b: UCMERCED LAND-USE DATASET
This publicly available dataset [33] is composed of 2,100 aerial images, each one with 256×256 pixels and 0.3-meter resolution per pixel. These images, obtained from different US locations, were classified into 21 classes: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium density residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis courts. As can be noticed, this dataset has highly overlapping classes such as the dense, medium, and sparse residential classes which mainly differs in the density of structures. Samples of these and other classes are shown in Figure 5.

FIGURE 4. The proposed morphological network DeepMorphNet conceived inspired by the AlexNet [2]. Since each layer is composed of distinct types of morphological neurons, the number of each type of neuron in each layer is presented as an integer times the symbol that represents that neuron (as presented in Section III-B). Hence, the total number of processing units in a layer is the sum of all neurons independently of the type. Also, observe that the two depthwise convolutions of a same layer share the kernel size, differing only in the stride and padding. These both parameters are presented as follows: the value related to the first depthwise convolution is reported separated by a comma of the value related to the second depthwise convolution. Although not visually represented, the pointwise convolutions explored in the morphological layers always use the same configuration: kernel 1×1, stride 1, and no padding.

FIGURE 5. Examples of the UCMerced Land-Use dataset.

FIGURE 6. Examples of the WHU-RS19 dataset.

c: WHU-RS19 DATASET

This public dataset [34] contains 1,005 high-resolution images with 600×600 pixels divided into 19 classes (approximately 50 images per class), including: airport, beach, bridge, river, forest, meadow, pond, parking, port, viaduct, residential area, industrial area, commercial area, desert, farmland, football field, mountain, park and railway station. Exported from Google Earth, that provides high-resolution satellite images up to half a meter, this dataset has samples collected from different regions all around the world, which increases its diversity but creates challenges due to the changes in resolution, scale, and orientation of the images. Figure 6 presents examples of some classes.

2) PIXEL CLASSIFICATION DATASETS

a: PAVIA CENTRE

This publicly available dataset [35] is composed of one image, acquired by the Reflective Optics System Imaging Spectrometer (ROSIS), covering the city of Pavia, southern Italy. This hyperspectral image has 715×1.096 pixels, spatial resolution of 1.3m per pixel, and 102 spectral bands. Pixels of this image are categorized into 9 classes. The false-color and ground-truth images, as well as the number of pixels in each class, are presented in Figure 7.
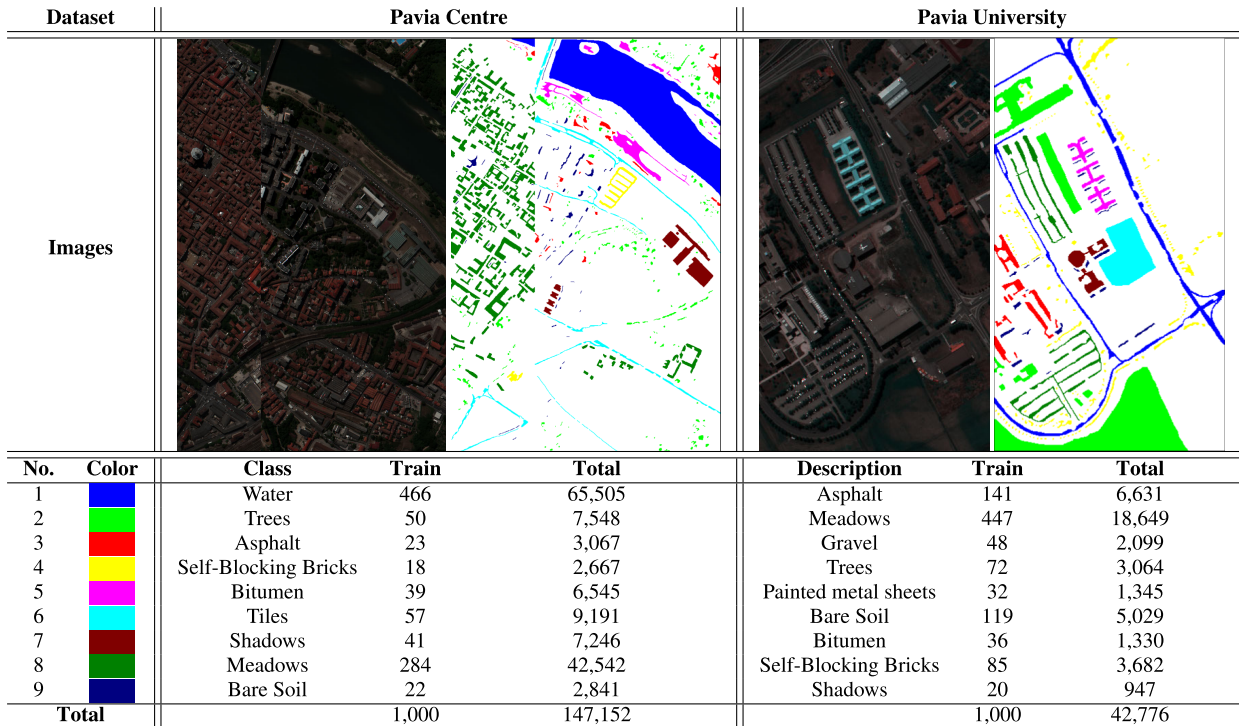
b: PAVIA UNIVERSITY

This public dataset [35] was also acquired by the ROSIS sensor during a flight campaign over Pavia. This hyperspectral image has 610×340 pixels, spatial resolution of 1.3m per pixel, and 103 spectral bands. Pixels of this image are also categorized into 9 classes. The false-color and ground-truth images, as well as the number of pixels in each class, are presented in Figure 7.

For both datasets, in order to reduce the computational complexity, Principal Component Analysis (PCA) [36] was used as a pre-processing method to reduce the dimensionality. Specifically, following [37], [38], for both datasets, we selected the first 3 principal components, which explain, approximately, 99% of the total variance of the data.

B. BASELINES

Several baselines were employed in this work. The first one, exploited in both tasks and all aforementioned datasets, is a standard convolutional version of the corresponding DeepMorphNet. This baseline, referenced as ConvNet, recreates the exact morphological architecture using the traditional convolutional layer but preserving all remaining configurations (such as filter sizes, padding, stride, etc). Moreover, this baseline makes use of max-pooling layers between the

Dataset			Pavia Centre			Pavia University		
Images								
No.	**Color**	**Class**	**Train**	**Total**	**Description**	**Train**	**Total**	
1		Water	466	65,505	Asphalt	141	6,631	
2		Trees	50	7,548	Meadows	447	18,649	
3		Asphalt	23	3,067	Gravel	48	2,099	
4		Self-Blocking Bricks	18	2,667	Trees	72	3,064	
5		Bitumen	39	6,545	Painted metal sheets	32	1,345	
6		Tiles	57	9,191	Bare Soil	119	5,029	
7		Shadows	41	7,246	Bitumen	36	1,330	
8		Meadows	284	42,542	Self-Blocking Bricks	85	3,682	
9		Bare Soil	22	2,841	Shadows	20	947	
Total			1,000	147,152		1,000	42,776	

FIGURE 7. In the first (top) part, false-color and ground-truth images of the explored hyperspectral datasets. In the second (bottom) part, the classes, the number of pixels used for training the models, and the total amount of labeled pixels.

convolutions, which makes it very similar to the traditional architectures of the literature [2]. An extension of this first baseline, referenced as PreMorph-ConvNet and exploited only for the (non-synthetic) image classification datasets, uses pre-defined morphological operations as pre-processing to extract the first set of discriminative features [20], [28]. Such data is then used as input to a ConvNet (in this case, the previously described one), which is responsible for performing the final classification. The third baseline, used only for the (non-synthetic) image classification datasets and referenced hereafter Depth-ConvNet, is exactly the Deep-MorphNet architecture but without using binary weights and depthwise pooling. This baseline reproduces the same DeepMorphNet architecture (also using depthwise and point-wise convolutions) and, consequently, **has the same number of parameters**, except for the binary weights. Other baselines, used only for the (non-synthetic) image classification datasets, are the deep morphological frameworks proposed by [23], [24]. In these cases, the DeepMorphNet architecture was recreated using both techniques. Finally, the last baseline, referenced with the suffix "SK" and explored only for the (non-synthetic) image classification datasets, is the convolutional and morphological networks, but with Selective Kernels (SK) [32], which allows the network to give more importance to certain feature maps than to others.

Differently from the morphological networks, all baselines use batch normalization [1] (after each convolution) and Rectified Linear Units (ReLUs) [30] as activation functions. It is important to note that, despite the differences, all baselines have the exact same number of layers and feature maps of the base DeepMorphNet. We believe that this conservation allows a fair comparison between the models given that the potential representation of the networks is somehow the same.

C. EXPERIMENTAL PROTOCOL

For the synthetic datasets, a train/validation/test protocol was employed. In this case, 60% of the instances were used as training, 20% as validation, and another 20% as test. Results of this protocol are reported in terms of the average accuracy of the test set. For the other image classification datasets, five-fold cross-validation was conducted to assess the accuracy of the proposed algorithm. The final results are the mean of the average accuracy (for the test set) of the five runs followed by its corresponding standard deviation. Finally, for the pixel classification datasets, following previous works [38]–[40], we performed a random sampling to select 1,000 training samples/pixels from all classes, as presented in (the bottom part of) Figure 7. All remaining pixels are used to assess the tested architectures. The final results are reported in terms of the average accuracy of those remaining pixels.

All networks proposed in this work were implemented using PyTorch. All experiments were performed on a 64 bit Intel i7 5930K machine with 3.5GHz of clock, Ubuntu 18.04.1 LTS, 64GB of RAM memory, and a GeForce GTX Titan X Pascal with 12GB of memory under a 10.0 CUDA version.

V. RESULTS AND DISCUSSION

In this section, we present and discuss the outcomes. Section V-A presents the results of the synthetic datasets while Section V-B discusses the results of the other datasets. Finally, Section V-C presents the results of the pixel classification datasets. Please remember that we present a proof of concept of a new paradigm: using non-linear morphological operations instead of standard linear convolutions for a deep network. To demonstrate the benefits of this proposal, similar architectures must be considered, including in terms of the number of neurons, layers, etc. In order not to confuse the benefits of proposal with the advantages of complex networks, as introduced, the proposed models were inspired by the simple AlexNet [2] architecture.

A. SYNTHETIC IMAGE CLASSIFICATION DATASETS

As explained in Section IV-A1, two synthetic datasets were proposed in this work to validate the feature learning of the deep morphological networks. Furthermore, as introduced, both datasets can be perfectly classified using one opening with specific SEs. Supported by this, the proposed DeepMorphSynNet, composed of one opening neuron, can be used to validate the feature learning process of the proposed technique, given that this network has the capacity of perfectly classifying the datasets as long as it successfully learns the SE.
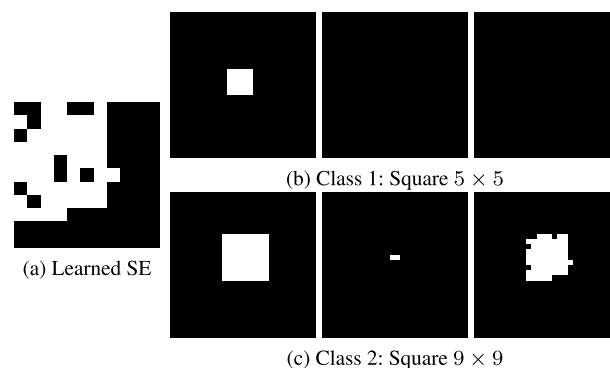
Given the simplicity of these datasets, aside from the first method describe in Section IV-B, we also employed as baseline a basic architecture composed uniquely of a classification layer. Specifically, this network has one layer that receives a linearized version of the input data and outputs the classification. The proposed morphological network and baselines were tested for both synthetic datasets using the same configuration, i.e., learning rate, weight decay, momentum, and number of epochs of 0.01, 0.0005, 0.9, and 10, respectively.

TABLE 1. Results, in terms of average accuracy, of the proposed method and the baselines for the synthetic datasets.

Dataset	Method	Average Accuracy (%)
Square	Classification Layer	86.50
	ConvNet	58.00
	DeepMorphSynNet	**100.00**
Rectangle	Classification Layer	52.00
	ConvNet	**100.00**
	DeepMorphSynNet	**100.00**

Results for the synthetic **square** dataset are presented in Table 1. Among the baselines, the worst results were generated by the ConvNets while the best outcome was produced by the network composed of a single classification layer (86.50%). A reason for this is that the proposed dataset does not have much visual information to be extracted by the convolution layer. Hence, in this case, the raw pixels themselves are able to provide relevant information for the classification. However, the proposed morphological network was able to outperform all baselines. Precisely, the DeepMorphSynNet yielded a 100% of average accuracy, perfectly classifying the entire test set of this synthetic dataset. As introduced in Section IV-A1, in order to achieve this perfect classification, the opening would require a square SE larger than 5×5 but smaller than 9×9 pixels. As presented in Figure 8a, this was the SE learned by the network. Moreover, as introduced, with this SE, the opening would erode the small 5×5 squares while keeping the larger 9×9 ones, the exact outcome of the morphological network, as presented in Figures 8b and 8c.
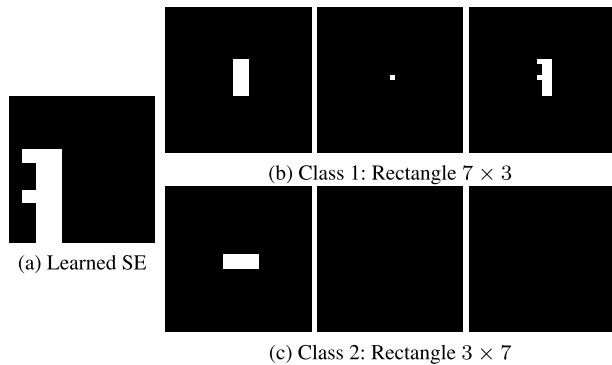

FIGURE 8. Visual results for the square synthetic dataset. (a) The learned SE. (b)-(c) Examples of the output of the opening neuron for two classes of the square synthetic dataset. The first column represents the input image, the second one is the output of the erosion, and the last one is the output of the dilation. Since erosion and dilation have tied weights (i.e., the same SE), they implement an opening.

Results for the synthetic **rectangle** dataset are presented in Table 1. In this case, the proposed DeepMorphSynNet and the ConvNet baseline produced perfect results, with the classification layer producing the worst result. Such results may be justified based on properties of this dataset: the distinct orientation of the rectangles may help both DeepMorphSynNet and ConvNet, while the exact same number of pixels for both rectangle classes may hinder the classification layer. As introduced in Section IV-A1, to perform this perfect classification, the opening operation (of the DeepMorphSynNet) would require a specific SE that should have the same orientation of one of the rectangles. As presented in Figure 9a, this is the SE learned by the morphological network. With such filter, the opening operation would erode the one type of rectangles while keeping the other, the exact outcome presented in Figures 9b and 9c.

B. IMAGE CLASSIFICATION DATASETS

For the image classification datasets, all networks were tested using essentially the same configuration, i.e., batch size, learning rate, weight decay, momentum, and number of epochs of 16, 0.01, 0.0005, 0.9, and 2,000 respectively. Aside from those approaches, we also employed as baseline an approach, called hereafter **Static SEs**, that reproduces the exactly morphological architecture but with static (non-optimized) SEs. In this case, each neuron has the configuration based on the most common SEs (mentioned

(a) Learned SE

(b) Class 1: Rectangle 7 × 3

(c) Class 2: Rectangle 3 × 7

FIGURE 9. Visual results for the rectangle synthetic dataset. (a) The learned SE. (b)-(c) Examples of the output of the opening neuron for two classes of the square synthetic dataset. The first column represents the input image, the second one is the output of the erosion, and the last one is the output of the dilation. Since erosion and dilation have tied weights (i.e., the same SE), they implement an opening.

in Section II-A). The features extracted by these static neurons are classified by a Support Vector Machine (SVM). The idea behind this baseline is to have a lower bound for the morphological network, given that this proposed approach should be able to learn better SEs and produce superior results.

TABLE 2. Results, in terms of accuracy, of the proposed method and the baselines for the image classification datasets.

Dataset	Method	Average Accuracy (%)	Number of Parameters (millions)	Training Time (hours per fold)
UCMerced	Static SEs	28.21±2.64	-	-
	ConvNet [2]	74.10±2.06	6.50	8.5
	PreMorph-ConvNet [20]	74.31±1.36	6.50	8.5
	Depth-ConvNet	75.24±2.17	7.47	101.1
	MorphoN [23]	34.52±4.93	0.31	1.0
	DMNN [24]	54.81±6.08	0.50	4.5
	DeepMorphNet (ours)	76.68±2.41	10.50	209.9
	ConvNet-SK [32]	75.05±2.26	9.36	12.1
	DeepMorph-SK (ours)	76.96±2.70	13.37	251.6
WHU-RS19	Static SEs	25.33±2.95	-	-
	ConvNet [2]	64.38±2.93	6.50	4.7
	PreMorph-ConvNet [20]	66.35±2.91	6.50	8.5
	Depth-ConvNet	65.25±2.12	7.47	44.7
	MorphoN [23]	40.88±2.60	0.31	0.5
	DMNN [24]	56.31±1.26	0.50	3.7
	DeepMorphNet (ours)	67.20±2.75	10.50	99.8
	ConvNet-SK [32]	72.06±2.27	9.36	6.6
	DeepMorphNet-SK (ours)	74.80±2.36	13.37	123.1

1) UCMerced LAND-USE DATASET

Results for the UCMerced Land-use dataset are reported in Table 2. In this case, all networks outperformed the lower bound result, generated by the Static SEs. Additionally, the MorphoN [23] and DMNN [24] networks were outperformed by all other approaches. In fact, all other experimented techniques, including the DeepMorphNets and the networks with Selective Kernels [32], yielded very similar results. This indicates that the proposed method is capable of optimizing the morphological filters to extract salient and relevant features, generating satisfactory outcomes. In order to grasp the difference between the information captured by

DeepMorphNets and ConvNets, we performed a comparison, presented in Figure 10, between the feature maps of such networks. As can be observed, there is a clear difference between the characteristics learned by the distinct networks. In general, the DeepMorphNet is able to preserve different features when compared to the ConvNets, which corroborates with our initial analysis.

To better evaluate the proposed morphological network, a convergence analysis of the architectures in the UCMerced Land-use dataset is presented in Figure 11a. For simplicity, only the networks that do not use selective kernels [32] were reported. Note that the proposed model is slower to converge when compared to the other networks. A reason for that is the large number of trainable parameters of DeepMorphNets, as presented in Table 2. However, given enough training time, all networks converge very similarly, which confirms that the proposed DeepMorphNets are able to effectively learn interesting SEs and converge to a suitable solution.

2) WHU-RS19 DATASET

The second part of Table 2 presents the results related to the WHU-RS19 dataset. Again, as expected, all architectures outperformed the lower bound result, generated by the Static SEs. As for the previous dataset, all experimented techniques, including the DeepMorphNets and the networks with selective kernels [32], achieved very similar outcomes, outperforming the MorphoN [23] and DMNN [24] approaches. These results reaffirm the previous conclusions related to the ability of the morphological networks to capture interesting features. Figure 10 presents the same comparison, as before, between the feature maps extracted from DeepMorphNets and ConvNets. Again, it is remarkable the difference between the features extracted by the distinct architectures, which corroborates with previous analysis and results.
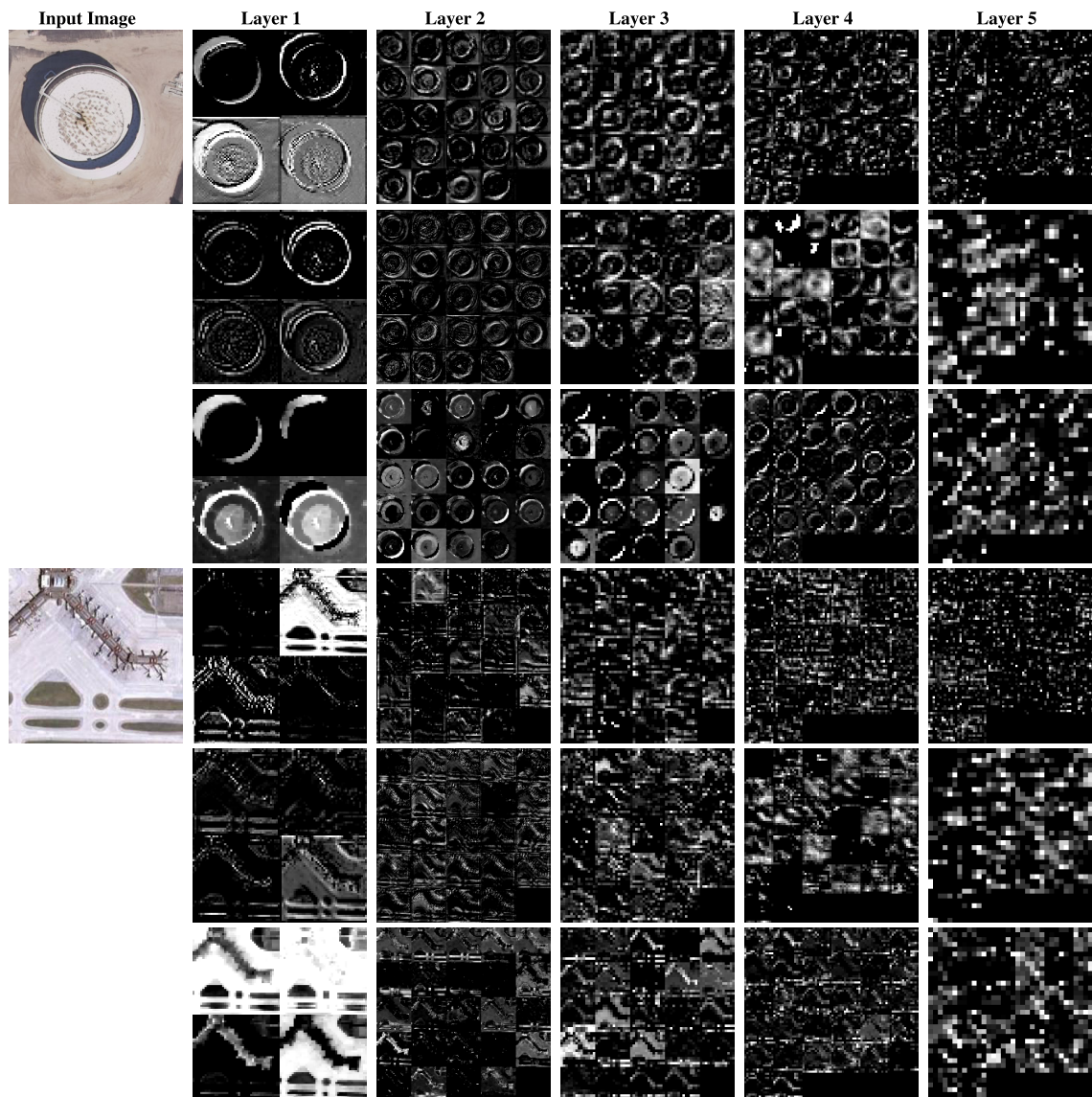
As for the previous dataset, a convergence analysis of the architectures in the WHU-RS19 dataset is presented in Figure 11b. Again, only the networks that do not exploit selective kernels [32] were reported. As before, the proposed model is slower (due to the number of trainable parameters), but able to converge if enough time is provided for the training.
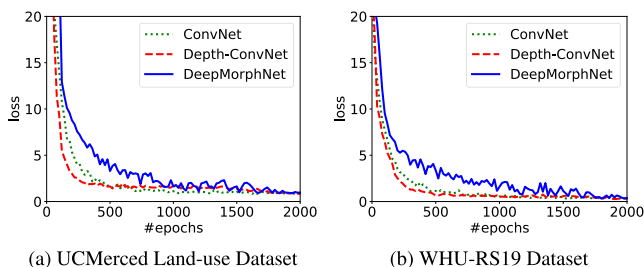
C. PIXEL CLASSIFICATION DATASETS

For the pixel classification datasets, all networks and baselines were assessed using the same configuration, i.e., batch size, learning rate, weight decay, momentum, and number of epochs of 10, 0.01, 0.0005, 0.9, and 1,000 respectively.

1) PAVIA CENTRE DATASET

Results for the Pavia Centre dataset are reported in (the first part of) Table 3. As can be seen through this table, the proposed DeepMorphNet outperformed the ConvNet model by almost 5 percentage points of average accuracy (88.28% versus 83.73%, respectively). This outcome shows, once again, that the proposed technique is capable of effectively learning relevant characteristics of the data.

FIGURE 10. Input images and some produced (upsampled) feature maps extracted from all layers of the networks for the UCMerced Land-use (top example) and RS19 Datasets (bottom example). Again, for simplicity, only the feature maps of the ConvNet (first row), Depth-ConvNet (second row), and DeepMorphNet (last row) models, without Selective Kernels [32], were reported.
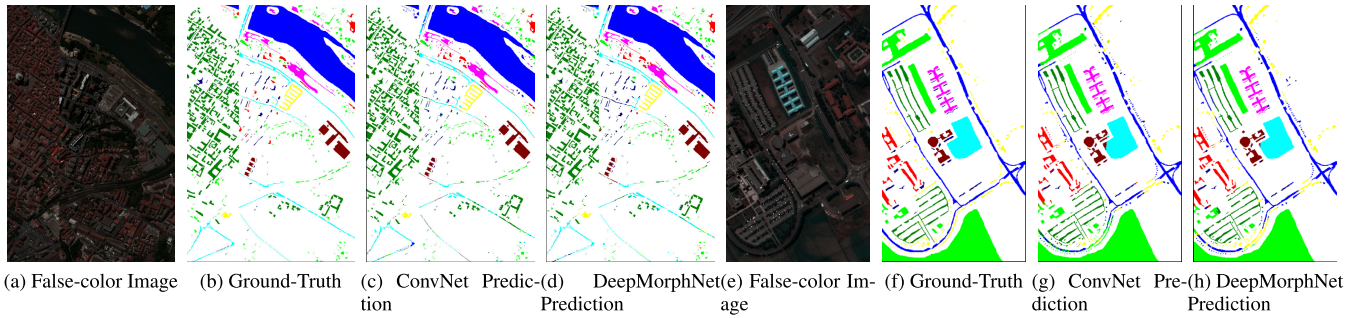

FIGURE 11. Convergence (fold 1) of the architectures for both datasets. Note that, for simplicity, only the networks without Selective Kernels [32] were reported.

In order to allow a visual comparison, Figure 12 presents the false-color image, the ground-truth, and the prediction maps generated by the proposed approach and the base-line. From this image, it is possible to observe that the
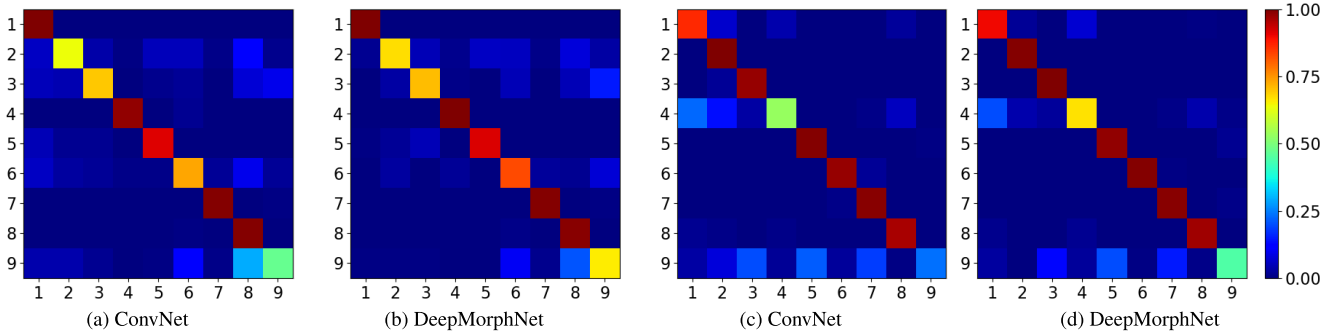
TABLE 3. Results, in terms of accuracy, of the proposed method and the baselines for the pixel classification datasets.

Dataset	Method	Average Accuracy (%)	Number of Parameters (millions)	Training Time (hours per fold)
Pavia Centre	ConvNet [2]	83.73	6.50	0.5
	DeepMorphNet (ours)	88.28	10.50	3.5
Pavia University	ConvNet [2]	82.44	6.50	0.5
	DeepMorphNet (ours)	86.52	10.50	3.5

DeepMorphNet produced a more consistent prediction map than the ConvNet baseline. We argue that this is mainly due to the fact that the proposed method can learn distinct and relevant patterns of specific classes, such as Tiles and Bare Soil (cyan and dark blue colors, respectively), an outcome that can be better perceived in the normalized confusion matrices presented in Figure 13.

| (a) False-color Image | (b) Ground-Truth | (c) ConvNet Prediction | (d) DeepMorphNet Prediction | (e) False-color Image | (f) Ground-Truth | (g) ConvNet Prediction | (h) DeepMorphNet Prediction |

FIGURE 12. The pixel classification datasets and the prediction maps generated by the proposed algorithm and baselines. Images (a)-(d) are related to the Pavia Centre dataset whereas images (e)-(h) are associated to the Pavia University dataset. The color encoding of the classes is presented in Figure 7.

| (a) ConvNet | (b) DeepMorphNet | (c) ConvNet | (d) DeepMorphNet |

FIGURE 13. Normalized Confusion matrices of the proposed method and baseline for both hyperspectral pixel classification datasets. Images (a) and (b) are related to the Pavia Centre dataset whereas images (c)-(d) are associated to the Pavia University dataset.

2) PAVIA UNIVERSITY DATASET

Results for the Pavia University dataset are reported in (the second part of) Table 3. Again, the proposed DeepMorphNet outperformed the ConvNet model by approximately 5 percentage points of average accuracy (86.52% versus 82.44%, respectively). This gain can be also seen in the prediction maps (Figure 12). Overall, the proposed method was capable of producing better results for some specific classes, such as Trees and Shadows (yellow and dark blue colors, respectively), an outcome that can be better observed in Figure 13.

VI. CONCLUSION

In this proof of concept work, we proposed a new paradigm for deep networks where linear convolutions are replaced by non-linear morphological operations. The proposed method, called Deep Morphological Network (DeepMorphNet), is able to perform morphological operations while optimizing their structuring elements. Technically, the proposed approach is composed of morphological layers, which consist of morphological neurons. Such processing units are built upon a framework that is essentially based on depthwise convolution and pooling layers. In fact, this framework provides support for the creation of the basic morphological neurons that perform erosion and dilation. These, in turn, allow the creation of other more complex ones that perform opening,

closing, top-hats, and (an approximation of) reconstructions. The proposed approach is trained end-to-end using standard algorithms employed in deep networks.

Experiments were conducted using **six** datasets: two synthetic, two image, and two pixel classification ones. The first two were only employed to analyze the feature learning of the proposed technique whereas the other datasets were employed to assess the efficiency of the DeepMorphNets. Results over the synthetic datasets have shown that the proposed DeepMorphNets are able to learn relevant structuring elements perfectly classifying them. Considering the image classification datasets, the proposed DeepMorphNets outperform other similar deep morphological frameworks [23], [24] whereas producing competitive results when compared to ConvNets with equivalent architectures. Finally, for the pixel classification datasets, the proposed method outperformed (in approximately 5 percentage points of average accuracy) standard convolutional networks. In general, a conclusion common to all datasets is that the morphological networks are capable of learning relevant filters and extracting salient features.

Although the presented conclusions may open opportunities towards a better use and optimization of non-linear morphological operations, learning the optimal shapes of the structuring elements used in each morphological neuron raises multiple scientific and technical challenges, including

the high number of trainable parameters (which increase the complexity of the network), and so on. Following the proof of concept presented in this manuscript, we will continue our work trying to tackle these computational limitations in order to consider more complex architectures. Furthermore, we plan to analyze the combination of DeepMorphNets and ConvNets and to test DeepMorphNets in different applications.

REFERENCES

[1] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep Learning*, vol. 1. Cambridge, MA, USA: MIT Press, 2016.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[3] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.

[4] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5012–5024, Nov. 2016.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[6] H. Lee and H. Kwon, "Going deeper with contextual CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 4843–4855, Oct. 2017.

[7] M. Zhang, W. Li, and Q. Du, "Diverse region-based CNN for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, Jun. 2018.

[8] K. Nogueira, M. D. Mura, J. Chanussot, W. R. Schwartz, and J. A. dos Santos, "Dynamic multicontext segmentation of remote sensing images based on convolutional networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7503–7520, Oct. 2019.

[9] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[10] R. Szeliski, *Computer Vision Algorithms and Applications*. London, U.K.: Springer, 2011.

[11] M. C. Motwani, M. C. Gadiya, R. C. Motwani, and F. C. Harris, "Survey of image denoising techniques," in *Proc. GSPX*, 2004, pp. 27–30.

[12] M. A. Oskoei and H. Hu, "A survey on edge detection methods," Univ. Essex, Colchester, U.K., Tech. Rep. CES-506, 2010.

[13] J. Serra and P. Soille, *Mathematical Morphology and Its Applications to Image Processing*, vol. 2. Berlin, Germany: Springer, 2012.

[14] M. Fauvel, J. Chanussot, J. A. Benediktsson, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2007, pp. 4834–4837.

[15] J. Xia, M. D. Mura, J. Chanussot, P. Du, and X. He, "Random subspace ensembles for hyperspectral image classification with extended morphological attribute profiles," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 4768–4786, Sep. 2015.

[16] Y. Kimori, K. Hikino, M. Nishimura, and S. Mano, "Quantifying morphological features of actin cytoskeletal filaments in plant cells based on mathematical morphology," *J. Theor. Biol.*, vol. 389, pp. 123–131, Jan. 2016.

[17] Y. Seo, B. Park, S.-C. Yoon, K. C. Lawrence, and G. R. Gamble, "Morphological image analysis for foodborne bacteria classification," *Trans. Amer. Soc. Agricult. Biol. Eng.*, vol. 61, no. 1, pp. 5–13, 2018.

[18] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2016, *arXiv:1610.02357*. [Online]. Available: http://arxiv.org/abs/1610.02357

[19] T. Liu, Y. Gu, J. Chanussot, and M. D. Mura, "Multimorphological superpixel model for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 6950–6963, Dec. 2017.

[20] A. Wang, X. He, P. Ghamisi, and Y. Chen, "LiDAR data classification using morphological profiles and convolutional neural networks," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 774–778, May 2018.

[21] Z. Yu-Qian, G. Wei-Hua, C. Zhen-Cheng, T. Jing-Tian, and L. Ling-Yun, "Medical images edge detection based on mathematical morphology," in *Proc. IEEE Eng. Med. Biol. 27th Annu. Conf.*, Jan. 2005, pp. 6492–6495.

[22] M. S. Miri and A. Mahloojifar, "Retinal image analysis using curvelet transform and multistructure elements morphology by reconstruction," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 5, pp. 1183–1192, May 2011.

[23] R. Mondal, M. S. Dey, and B. Chanda, "Image restoration by learning morphological opening-closing network," *Math. Morphol.-Theory Appl.*, vol. 4, no. 1, pp. 87–107, Jan. 2020.

[24] G. Franchi, A. Fehri, and A. Yao, "Deep morphological networks," *Pattern Recognit.*, vol. 102, Jun. 2020, Art. no. 107246.

[25] J. Masci, J. Angulo, and J. Schmidhuber, "A learning framework for morphological operators using counter–harmonic mean," in *Proc. Int. Symp. Math. Morphol. Appl. Signal Image Process.* Berlin, Germany: Springer, 2013, pp. 329–340.

[26] D. Mellouli, T. M. Hamdani, M. B. Ayed, and A. M. Alimi, "Morph-CNN: A morphological convolutional neural network for image classification," in *Proc. Int. Conf. Neural Inf. Process.* Cham, Switzerland: Springer, 2017, pp. 110–117.

[27] S. R. Borra, G. J. Reddy, and E. S. Reddy, "Classification of fingerprint images with the aid of morphological operation and AGNN classifier," *Appl. Comput. Informat.*, vol. 14, no. 2, pp. 166–176, Jul. 2018.

[28] E. Aptoula, M. C. Ozdemir, and B. Yanikoglu, "Deep learning with attribute profiles for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1970–1974, Dec. 2016.

[29] P. Kalshetti, M. Bundele, P. Rahangdale, D. Jangra, C. Chattopadhyay, G. Harit, and A. Elhence, "An interactive medical image segmentation framework using iterative refinement," *Comput. Biol. Med.*, vol. 83, pp. 22–33, Apr. 2017.

[30] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[31] K. Nogueira, W. O. Miranda, and J. A. Dos Santos, "Improving spatial feature representation from aerial scenes by using convolutional networks," in *Proc. 28th SIBGRAPI Conf. Graph., Patterns Images (SIBGRAPI)*, Aug. 2015, pp. 289–296.

[32] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[33] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst. (GIS)*, 2010, pp. 270–279.

[34] G.-S. Xia, W. Yang, J. Delon, Y. Gousseau, H. Sun, and H. Maître, "Structural high-resolution satellite image indexing," in *Proc. 7th Symp.-Years ISPRS TC*, vol. 38, 2010, pp. 298–303.

[35] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, pp. S110–S122, Sep. 2009.

[36] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*. Boston, MA, USA: Cengage Learning, 2014.

[37] L. Jiao, M. Liang, H. Chen, S. Yang, H. Liu, and X. Cao, "Deep fully convolutional network-based spatial distribution prediction for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5585–5599, Oct. 2017.

[38] W. Hu, H. Li, L. Pan, W. Li, R. Tao, and Q. Du, "Spatial–spectral feature extraction via deep ConvLSTM neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 6, pp. 4237–4250, Jun. 2020.

[39] X. Ji, Y. Cui, H. Wang, L. Teng, L. Wang, and L. Wang, "Semisupervised hyperspectral image classification using spatial–spectral information and landscape features," *IEEE Access*, vol. 7, pp. 146675–146692, 2019.

[40] J. Feng, J. Chen, L. Liu, X. Cao, X. Zhang, L. Jiao, and T. Yu, "CNN-based multilayer spatial–spectral feature fusion and sample augmentation with local and nonlocal constraints for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 4, pp. 1299–1313, Apr. 2019.

KEILLER NOGUEIRA received the B.Sc. degree in computer science from the Universidade Federal de Viçosa, Brazil, in 2012, and the M.Sc. and Ph.D. degrees in computer science from the Universidade Federal de Minas Gerais, Brazil, in 2015 and 2019, respectively. He is currently a Lecturer with the Division of Computing Science and Mathematics, University of Stirling, U.K. He has published several high-quality articles in leading journals and conferences. His research interests include deep and machine learning, pattern recognition, image processing, computer vision, and remote sensing.

JOCELYN CHANUSSOT (Fellow, IEEE) received the M.Sc. degree in electrical engineering from Grenoble Institute of Technology (Grenoble INP), Grenoble, France, in 1995, and the Ph.D. degree from the Université de Savoie, Annecy, France, in 1998.

He was with the Geography Imagery Perception Laboratory for the Delegation Generale de l'Armement (DGA - French National Defense Department), in 1999. Since 1999, he has been with Grenoble INP, where he is currently a Professor of signal and image processing. He has been a Visiting Scholar at Stanford University (USA), KTH (Sweden) and NUS (Singapore). Since 2013, he has been an Adjunct Professor with the University of Iceland. From 2015 to 2017, he was a Visiting Professor with the University of California, Los Angeles (UCLA). He is conducting his research at Grenoble Images Speech Signals and Automatics Laboratory (GIPSA-Lab). His research interests include image analysis, multicomponent image processing, nonlinear filtering, data fusion, and machine learning in remote sensing. He is a member of the IEEE Geoscience and Remote Sensing Society AdCom. He is a member of the Institut Universitaire de France (2012–2017). He is the Founding President of the IEEE Geoscience and Remote Sensing French Chapter (2007–2010) which received the 2010 IEEE GRS-S Chapter Excellence Award. He was a co-recipient of the NORSIG 2006 Best Student Paper Award, the IEEE GRSS 2011 and 2015 Symposium Best Paper Award, the IEEE GRSS 2012 Transactions Prize Paper Award, and the IEEE GRSS 2013 Highest Impact Paper Award. He was the General Chair of the first IEEE GRSS Workshop on Hyperspectral Image and Signal Processing, Evolution in Remote Sensing (WHISPERS). He was the Chair (2009–2011) and the Co-Chair of the GRS Data Fusion Technical Committee (2005–2008). He was a member of the Machine Learning for Signal Processing Technical Committee of the IEEE Signal Processing Society (2006–2008) and the Program Chair of the IEEE International Workshop on Machine Learning for Signal Processing, in 2009. He is an Associate Editor of the IEEE Transactions on Geoscience and Remote Sensing and the IEEE Transactions on Image Processing. He was the Editor-in-Chief of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (2011–2015). He was a Guest Editor of the Proceedings of the IEEE, in 2013, and the *IEEE Signal Processing Magazine*, in 2014. He is a 2018 Highly Cited Researcher (Clarivate Analytics), College Station, TX, USA.

MAURO DALLA MURA (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in telecommunication engineering from the University of Trento, Italy, in 2005 and 2007, respectively, and the joint Ph.D. degree in information and communication technologies (telecommunications area) from the University of Trento, Italy, and in electrical and computer engineering from the University of Iceland, Iceland, in 2011. He was a Research Fellow with Fondazione Bruno Kessler, Trento, Italy, in 2011, conducting research on computer vision. Since 2012, he has been an Assistant Professor with Grenoble Institute of Technology (Grenoble INP), France. He is conducting his research at Grenoble Images Speech Signals and Automatics Laboratory (GIPSA-Lab). He has been appointed as a "Specially Appointed Associate Professor" with the School of Computing, Tokyo Institute of Technology, Japan, during 2019–2022. His research interests include remote sensing, image processing, pattern recognition, mathematical morphology, classification, and multivariate data analysis. He was a recipient of the IEEE GRSS Second Prize in the Student Paper Competition of the 2011 IEEE IGARSS 2011 and a co-recipient of the Best Paper Award of the *International Journal of Image and Data Fusion* for the year 2012–2013 and the Symposium Paper Award for IEEE IGARSS 2014. Since 2016, he has been the President of the IEEE GRSS French Chapter (he previously served as Secretary 2013–2016). In 2017, the IEEE GRSS French Chapter was a recipient of the IEEE GRSS Chapter Award and the "Chapter of the year 2017" from the IEEE French Section. Since 2016, he has been on the Editorial Board of the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (J-STARS).

JEFERSSON A. DOS SANTOS (Member, IEEE) received the Ph.D. degree in computer science from the Université de Cergy-Pontoise, France, and the University of Campinas, Brazil, in 2013. Since 2016, he has a Research Productivity Scholarship from the Brazilian Research Council (CNPq). He is currently an Associate Professor with the Department of Computer Science, Universidade Federal de Minas Gerais, Brazil. He has published several articles in journals with high-impact factor and selective editorial policy. He has published more than 30 articles in important conferences of remote sensing, image processing, and computer vision areas. He has experience in coordinating research with Brazilian funding agencies and research and development projects with companies in those topics. He is the Founder and the Coordinator of the Laboratory of Pattern Recognition and Earth Observation (PATREO-www.patreo.dcc.ufmg.br), one of Brazil's pioneer groups focused on the development of computer vision and machine learning for remote sensing applications.

● ● ●