DeepNEM: Deep Network Energy-Minimization for Agricultural Field Segmentation

Margarita Torre, Beatriz Remeseiro, Petia Radeva, *Fellow, IEEE*, and Fernando Martinez

***Abstract*—One of the main characteristics of agricultural fields is that the appearance of different crops and their growth status, in an aerial image, is varied, and has a wide range of radiometric values and high level of variability. The extraction of these fields and their monitoring are activities that require a high level of human intervention. In this article, we propose a novel automatic algorithm, named deep network energy-minimization (DeepNEM), to extract agricultural fields in aerial images. The model-guided process selects the most relevant image clues extracted by a deep network, completes them and finally generates regions that represent the agricultural fields under a minimization scheme. DeepNEM has been tested over a broad range of fields in terms of size, shape, and content. Different measures were used to compare the DeepNEM with other methods, and to prove that it represents an improved approach to achieve a high-quality segmentation of agricultural fields. Furthermore, this article also presents a new public dataset composed of 1200 images with their parcels boundaries annotations.**

***Index Terms*—Agricultural fields, image edge analysis, image segmentation, region extraction.**

I. Introduction

CONTINUOUS development of digital aerial images has provided constant improvement of spatial resolution. The availability of very high-resolution satellite images and unmanned aerial vehicles provide a cheaper and faster way to obtain detailed information for large areas. Attempting to locate and identify relevant elements as automatically as possible has been a research field in constant development. Agricultural field analysis has been handled from the very beginning as one of the first applications in image processing, due to agricultural analysis policies and taxes applied to agricultural land uses. Manual digitization is the most common way to acquire detailed information, making it a very time-consuming process whose results depend on the operator's interpretation. In order to overcome these challenges, there is a strong need for accurate, low cost, innovative methods for information extraction.

Most automatic remote sensing approaches today are based on clustering techniques [1], [2]. Despite hundreds of algorithms having been proposed for segmentation, only a few of them have been implemented and are available as a production tool [3]. Among them, eCognition[1] is the most popular and widely used segmentation software, converted into a productive software gold standard tool [1]. It is based on fuzzy segmentation allowing better retention of the radiometric variation of the agricultural regions. eCognition provides region contours by aggregation, which sometimes leads to oversegmentation, and thus, leaves room for improvement, mainly in postprocessing editing tasks. Other fuzzy c-means clustering approaches are found in [2], used together with spatial constraints. However, unsupervised clustering methods have several limitations—the number of clusters are often unknown and the predefined parameters often deliver over/undersegmented results, requiring further split and merge procedures in combination with an interpretation of the segmented images.

Edges have been another approach used in automatic and semiautomatic techniques for region boundary delineation [4]. However, these methods suffer the problems of detecting false edges, locating poor edges, and missing edges, which limit its applicability. A more recent agricultural boundary detection technique used by Alemu [5] is the line segment detection algorithm, aimed at detecting straight contours in images [6]. These types of methods perform reasonably well in regularly shaped agricultural fields, but fail when dealing with heterogeneous datasets. Additionally, they are generally sensitive to intraclass variability, thus leading to oversegmentation. When working with methods directed by models, the better the model represents the knowledge of scene interpretation, the more useful the extracted information will be. Energy-minimization (EM) theory delivers a common framework to unify different model-based approaches, such as graph-cuts [7], random walker [8], and shortest path [9]. One limitation of these approaches is that, in practice, the edge-stopping in the minimizing function is never

Manuscript received September 27, 2019; revised January 12, 2020; accepted January 27, 2020. Date of publication February 7, 2020; date of current version February 21, 2020. This work was supported in part by MINECO (Spanish Ministry of the Economy and Competitiveness) under Grant TIN2015-65069-C2-2-R, Grant TIN2017-89244-R, and Grant RTI2018-095232-B-C2 from the MINECO (Spanish Ministry of the Economy and Competitiveness); 2017SGR-01742 (CVUB) and CERCA Programme from the Generalitat de Catalunya; and in part by the Principado de Asturias Regional Government, partially supported with EDRF funds under Grant IDI-2018-000176. The work of P. Radeva was supported by ICREA Academia 2014. *(Corresponding author: Fernando Martinez.)*

Margarita Torre is with the Department of Computer Science, Universitat Autònoma de Barcelona, 08193 Bellaterra, Spain, and also with the Department of Territory and Sustainability, Generalitat de Catalunya, 08029 Barcelona, Spain (e-mail: margarita.torre@gencat.cat).

Beatriz Remeseiro is with the Department of Computer Science, Universidad de Oviedo, 33023 Gijón, Spain (e-mail: bremeseiro@uniovi.es).

Petia Radeva is with the Departament de Matemàtiques i Informàtica, Universitat de Barcelona, 08007 Barcelona, Spain, and also with the Computer Vision Center, 08193 Bellaterra, Spain (e-mail: petia.ivanova@ub.edu).

Fernando Martinez is with the Departament de Matemàtiques, Universitat Politècnica de Catalunya, 08034 Barcelona, Spain (e-mail: fernando.martinez@upc.edu).

This article has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

Digital Object Identifier 10.1109/JSTARS.2020.2971061

[1] Online. [Available]: http://www.ecognition.com/

exactly zero in the edges, and so the curve may eventually pass through object boundaries.

In recent years, deep learning approaches have been achieving popularity and impressive performance in detection, segmentation, and recognition of objects and regions in images [10]. Deep learning techniques, and more specifically convolutional neural networks, have also been applied in an important number of research works focused on edge detection. Among them, it is worth noted the holistically-nested edge detector (HED) [11], [12], an efficient and accurate network that performs image-to-image training and prediction. The proposed architecture connects its side output layers to the last convolution layer of each stage in a VGGNet [13]. Further works found in the literature focused their attention on HED, thus highlighting its reliability for edge detection. Liu and Lew [14] used relaxed labels generated by bottom-up edges to guide the training process of HED. Liu *et al.* [15] proposed an edge detector that uses different image scales and aspect ratios to learn rich hierarchical representations, with an architecture that only adds 1×1 convolutional layers to HED. The term *nested* is due to the inherited and progressively refined edge maps produced as side outputs, thus making successive edge maps more concise. The term *holistic*, despite not explicitly modeling the structured output, is because the network aims at training and predicting edges in an image-to-image fashion. More recently, [16] introduced a bidirectional cascade structure to enforce each layer (BDCN), which aims at focusing on a specific scale. However, deep learning techniques applied to agricultural field segmentation is a highly underexplored field of research.

In terms of segmenting land coverage elements, [3] showed a complete review of segmentation methods widely used in this context, whose application is not limited to urban coverage. To enlighten some combined approaches, [17] presented two proposals based on a multiparadigm collaborative framework—the first one is inspired by cascading techniques in machine learning, while the second one applies many collaborating one-vs-all class extractors in parallel. Csillik [18] proposed to use a partition delivered by the simple linear iterative clustering superpixel algorithm as a starting segmentation point. Gu *et al.* [19] presented another approach composed of a minimum spanning tree algorithm for the initial segmentation, and the minimum heterogeneity rule algorithm for object merging in a fractal net evolution approach. All these methods show that the segmentation of aerial images should be based on edges instead of regions, since different agricultural fields often can be composed of similar or even the same crop (see Fig. 1). For this reason, recent and powerful generic methods for semantic image segmentation based on neural networks [20] are unsuitable for aerial image segmentation.

Given the problem of locating and extracting agricultural fields in aerial images, the most relevant clues that assist users in detecting their boundaries are their regular shape and the fact that a field can be distinguished from its surroundings—since it is limited by linear elements or by other fields that do not follow its homogeneity or its texture pattern. As far as clues are concerned, edge extraction procedures play an important role in the problem at hand. For this reason, we propose taking advantage of the HED network [12], based on nested multiscale feature learning and deeply supervised networks [21]. HED is able to automatically learn the type of rich hierarchical features that are crucial in the process of disambiguation of natural aerial images and field boundary detection.

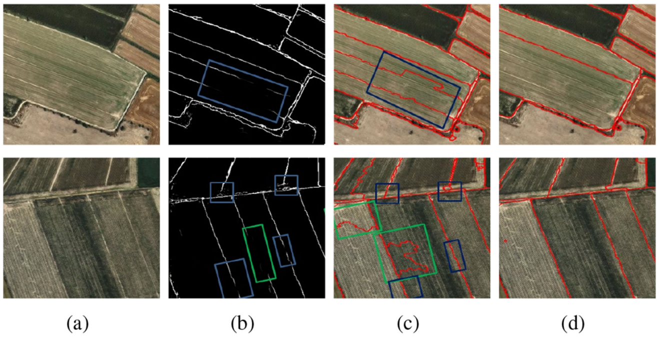

Fig. 1. (a) Two original images. (b) Edges delivered by HED (with gaps and isolated elements highlighted in squares). (c) Regions obtained with the EM process. (d) Our final solution after the model fitting step.

Deep networks (DNs), such as HED, are good candidates to fully extract an important portion of aerial field boundaries. However, some of them may not be detected, while other elements may be wrongly detected as boundaries. For this reason, [15] proposed refining the output provided by HED, using only the pixels with the greatest amount of annotators labeled as positive samples, due to their high consistency and their ease of training. When working with aerial images, gaps or isolated elements may also appear when applying edge detectors such as HED, as shown in Fig. 1(b). In this case, all the edge pixels obtained with HED must be considered, and properly selected depending on their reliability in model extent, thus making the approach proposed by [15] not entirely adequate. Although HED provides excellent results for image edge detection, it cannot ensure obtaining closed boundaries of image regions. To this aim, we propose to integrate the HED detector into an EM framework. The main idea is that the edges obtained with HED must be processed, categorized, and completed by the EM process to assure straightforward extraction of agricultural fields. Moreover, the boundaries added must be finally accepted or not depending on whether they follow sufficiently the tracks given by the edges delivered by HED. Fig. 1(c) and (d) shows how the EM step followed by a model fitting process are able to solve the problems detected when applying HED (i.e., gaps and isolated elements).

Summarizing, this work presents a new model called DeepNEM, as an automatic global segmentation approach that integrates HED as a DN for edge detection with an EM procedure. Our approach relies on boundary clues, which must follow a predefined model and are also used to complete the boundaries by an EM global process. Our main contributions include the following:

1) a novel method for aerial image segmentation named DeepNEM, which integrates a DN with an EM stage to refine and complete the clues delivered by the network, thus solving the lack of boundary information;

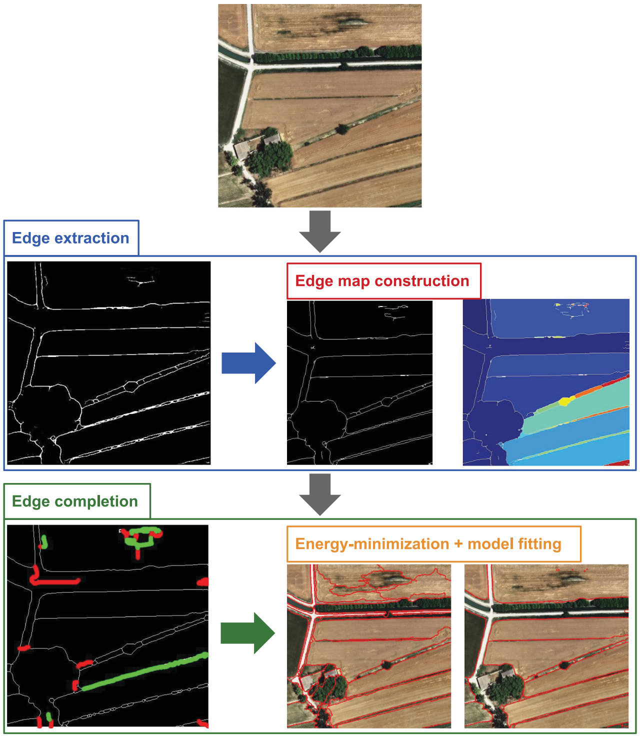

Fig. 2. Main steps of our DeepNEM approach for region segmentation of agricultural fields. The edge extraction step (second row) goes from the edges delivered by HED (left) to the selection of relevant ones (center), and the main regions formed by them (right). The edge completion (third row) is composed of the energy minimization step applied to the edges previously selected (left)—red lines are edges allowed to be modified—whose results are expanded under radiometric constraints (center), and the final output with the fields that fulfill the model (right).

2) an extensive validation of our method that includes different performance metrics and a comparison with state-of-the-art approaches;
3) a complete dataset composed by 1200 $(500 \times 500 \times 3)$ images and their corresponding parcel delineation annotations to serve as a benchmark for future research in the field of aerial image segmentation.

The rest of this article is organized as follows. Section II describes the DeepNEM method with its main steps, Section III presents the new dataset and the experimental results. Finally, Section IV closes with conclusions and future research.

II. Methodology

Discontinuity in terms of radiometry or texture homogeneity is the characteristic that catches the eye when operators delineate the field boundaries in aerial images. Since these interruptions are the common evidence among neighboring fields, we strongly rely on edges to drive and define the mainstream of the process. Our method is divided into two main steps, as shown in Fig. 2—edge extraction and edge completion. These main steps contain side refinements to reinforce some evidence and to complete or dismiss some clues. In particular, the edge extraction is completed with morphological operations to deliver an edge map, while the edge completion delivers enough cues to analyze if they are worth including in the final solution. Below, both steps are described in depth.

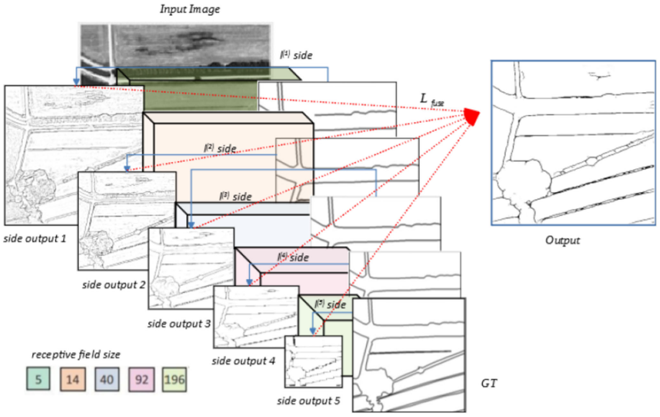

Fig. 3. Illustration of the HED architecture for edge detection over an aerial input image, highlighting the error back propagation paths. Note that side-output layers are inserted after convolutional layers, which are shown as colored boxes, with the side output plane size becoming smaller and the receptive field size becoming larger.

A. Edge Extraction: From Image to Edges

To recover discontinuities in radiometric or texture homogeneity of aerial images, we rely on HED [12] in order to locate regions of contrast changes, usually corresponding to different fields, and to maintain their singularities, such as wooded areas, trees, and high contrast linear elements. The HED architecture, as shown in Fig. 3, is composed of a single-stream DN with multiple side outputs, and uses a deep architecture to simulate the perceptual multilevel human approach. Notice that the structure in multiple stages with different strides is useful to capture the inherent scales of edge maps. There is also a weighted-fusion layer-error to help with the update of the output-layer parameters by back-propagation and to learn the fusion weight during training.

Fig. 3 shows the error back-propagation paths as well as the importance of inserting the output convolutional layers. For each side-output layer, deep supervision is imposed, guiding the side-outputs toward edge predictions with the desirable characteristics. Fig. 3 also shows that the outputs of HED are both multiscale and multilevel, as well as how the side-output-plane size becomes smaller while receptive field size becomes larger. Note that the entire network is trained with multiple error propagation paths (dashed lines).

Despite the high-quality edge detection by HED, it does not assure closed regions. For this reason, we rely on an EM model to guide the process that extracts from the edges as many clues as possible, moreover to complete them when it is necessary. Some linear elements detected by HED may reflect confusing clues, because they lie inside some regions (if kept, they will oversegment them) or near to hard edges, when undoubtedly reliable boundaries are extracted. For this reason, we propose a side refinement of this stage, guided by an EM model that keeps the knowledge of what real agricultural fields are like.

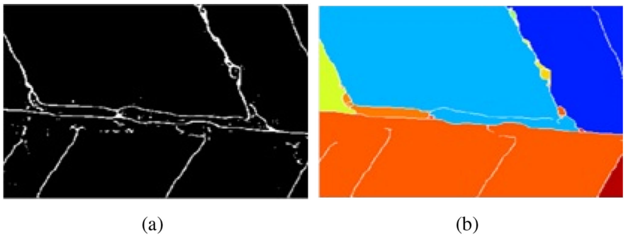
(a) (b)

Fig. 4. (a) Output edges obtained with HED, and (b) result in terms of *structuring edges*.

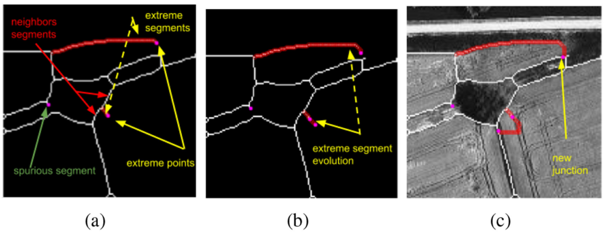

Fig. 5. (a) Different elements involved in the EM step, (b) their evolution throughout the process, and (c) the completed edges obtained at the end.

1) Edge Map Construction: Among all the edges detected by HED, it is necessary to select the ones that better suit an agricultural field segmentation, since they define a preliminary image division into the main regions. If we process all the edges in terms of linear elements, we will find that some of the selected segments will not deliver relevant clues (e.g., close parallel elements that cause closed elongated areas, or small gaps between long segments that the model tends to close). Since we want to detect areas, instead of acting directly on the linear evidence, we will always take into consideration the regions that edges form. This approach will handle boundary information, such as bushes, regardless of whether they divide regions or appear within closed regions, as well as other important elements (see Fig. 4).

In order to reinforce strong clues or to eliminate these potentially misleading clues, a morphological process is applied, which is composed of several serialized morphological operations to keep and merge adjacent small areas and to narrow boundaries. These operations are a sequence of opening, closing, and thinning morphological functions to filter profiles, without losing relevant evidence. This process is addressed by a predefined threshold, named $A_{\min}$, which represents the minimum size of the allowed artifacts.

B. Edge Completion: From Edges to Regions

At this point, the image is divided into regions whose boundaries are completely closed by edges, named tiles. For each one, the algorithm analyzes its content, when it has edges with gaps or isolated edges. The last ones are prone to becoming part of a boundary, while the others are analyzed to be completed. Despite the high quality of the edges provided by HED, the algorithm does not guarantee that edges would form closed contours. An additional step is necessary to complete clues and avoid undersegmentation problems.[2] Fig. 4 shows an example of boundaries with gaps and others with isolated clues.

1) Energy-Minimization: In order to complete the boundaries, we integrate the HED output into an EM process that obtains a complete first approximation of the boundary segmentation inside each tile. This process finds the shortest sequence of pixels between segments that have an extreme sequence. The sequence is obtained by forcing the total energy of the edges to be minimum. Not only does it take into consideration the nearest environment of each gap, but it also considers each edge within a broad area.

The tiles obtained in the previous stage, formed by a closed chain of long segments, will be the domain of the *edge completion* step. This process is driven by *relevant points*, namely X-connected points (those with X edge pixels among their eight closest neighbors) with $X \neq 2$, and divided into: *extreme points*, when $X = 1$; and *junctions*, when $X \geq 3$. For each *extreme point*, its associated chain of 2-connected points is considered a *segment*. Note that segments are divided depending on their length, defined by the parameter $T_{\min}$, into long and short segments. The different classes of long and short elements are shown in Fig. 5, and described below.

Long segments are classified into the following.

1) *Extreme segments:* Segments limited by at least one extreme, while the other end can be an extreme or a junction.
2) *Arcs:* Segments limited by two junctions.

Short segments are considered in the minimization process, but are forbidden from growing. They are classified as follows.

1) *Isles:* Segments limited by two *extreme* points. They will be taken into account in the filling process to attract the edges.
2) *Spurious:* Segments limited by an extreme point and a junction. In the junction, they join a single long segment, alongside other spurious ones that may also share the junction.

For each *extreme* segment, the EM will deliver a new end location at each step. This sequential iterative process will stop when, in the elongation process, another segment is reached. Depending on the type of element reached, the process will end by elongating both segments to a common point, or by creating a junction. Next, these elements whose length is under a threshold ($T_{\min}$) are deleted (short segments), while the other ones are kept (long segments).

The minimization result follows the desired model, which is a combination of smoothness and minimum length segments, where the relevant points are the *junctions* and *extremes*. For each *extreme* segment, a potential is defined by a force that rejects the *extreme* point—the n_i pixels that form the segment (x_i, y_i) try to expel the *extreme*—and a force that attracts this *extreme,* due to all other surrounding segments. So, for each *extreme* segment, its *extreme* point (x_{n_i}, y_{n_i}) will be newly located at the position that tries to minimize the functional

$$V_{n_i} = \sum_{i \in \text{edge}} \frac{1}{r_{i,n_i}} - \sum_{i \notin \text{edge}} \frac{1}{r_{i,n_i}}$$

[2] A key problem in segmentation is that of dividing a region into too few (undersegmentation) or too many areas (oversegmentation).

where $r_{i,n_i} = \sqrt{(x_i - x_{n_i})^2 + (y_i - y_{n_i})^2}$. Minimizing V_{n_i} is equivalent to solving[3]

$$\vec{F} = -\vec{\nabla} V = 0, \qquad \vec{\nabla} V = \left(\frac{\partial V}{\partial x}, \frac{\partial V}{\partial y}\right).$$

We propose a greedy method that at each step finds a new *extreme*, stopping when this *extreme* reaches another segment. If the segment reached is an *extreme* segment, it can be analyzed to become a corresponding element. Otherwise, the process will stop by reaching an *arc* segment and the growing process is frozen. Note that the *extreme* segment will be awakened if another *extreme* segment connects to it, during its growing process. In such a case, the segments will become corresponding segments. If the process is completed and no other *extreme* segment has been reached, a *junction* point in the *arc* segment is created. Fig. 5 shows an example of these elements.

The notation used in the detailed formulation is as follows.

1) For each *extreme* segment e^i, we describe its coordinates obtained from the edge extraction as $\vec{e}^{\,i}_j\ \{j = 1 : n_i\}$, where n_i is its length.
2) Associated with each element e^i, we code the *spurious* segments, such as s^{ij}, where $j \in \{1 : n\_esp_i\}$. The identification of segments is esp^i_l, where $\{l = 1 : n\_\text{esp}_i\}$. Fig. 5 includes an example of *extreme* and their spurious segments.
3) The *arcs* that connect directly with *extreme* segments (e^i) are called neighbors; being ngh_i the number of segments associated with e^i. The components of these segments are *prohibited* locations for the growing process in the first iterations, in order to avoid loops.
4) During the growing process, the *extreme* segment is allowed to reach another segment that is not from its neighbor. In this case, we consider that the segment touches an edge at the position $\vec{p}$ (bit on). $I(\vec{p})$ represents the edge image value (1 or 0) at $\vec{p}$.

In the energy formulation, for each *extreme* segment e^i, we consider that the rejecting force over the point $\vec{x}$, $f_R(e^i, \vec{x}) = \sum_{k=1}^{n_i} \frac{\vec{e}^{\,i}_k - \vec{x}}{\|\vec{e}^{\,i}_k - \vec{x}\|^3}$, comes from the *extreme* segment components—both the original and the added points, as well as the expelling force due to the associated spurious segments $\vec{s}^{\,ij}_k$ and the *neighbor* segments. Each one is weighted in different ways, depending on the confidence in the data. For example, the weight associated with the edge points ω_e is greater than the one associated with the added coordinates ω_{add}. The *spurious* segments linked to an *extreme* segment will produce the same effects as the edge points of the *extreme* segment. For that reason, the parameters ω_{esp} and ω_{ngh} normally have the same value as ω_e. As we can see, the point distance will reduce the magnitude of the force.

The subtractive contribution $f_A(e^i, \vec{x}) = \sum_{k=n_i+1}^{m_i} \frac{\vec{x} - \vec{e}^{\,i}_k}{\|\vec{x} - \vec{e}^{\,i}_k\|^3}$ carried by the attracting components, comes from the rest of the *arcs* not directly connected to the *extreme* segment e^i. We apply the same reasoning to the rejecting force, for the different elements—the original edge points are stronger than the added ones. We use ω_{biton} to weight the elements that come from the edge image map; and we take into account especially the points in a round environment $B(\cdot, r)$—in the examples we have chosen $r = 2 \cdot A_{\min}$, due to the fact that larger circular areas take more computational time and the results do not improve. From the final *extreme* position $\vec{e}^{\,i}_{m_i}$, the force delivers a new segment end position by generating a new component $\vec{e}^{\,i}_{m_i+1}$

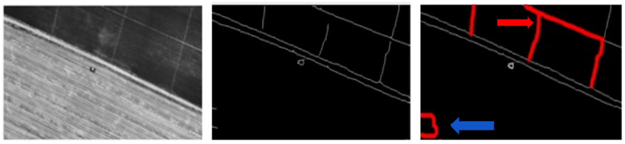
(a) (b) (c)

Fig. 6. (a) Input image, (b) edges obtained with HED, and (c) completed edges. The red arrow points to the accepted added boundaries, while the blue one points to the area rejected.

$$\begin{aligned}\vec{e}^{\,i}_{m_i+1} = {}& \vec{e}^{\,i}_{m_i} + \omega_e\, f_R(e^i, \vec{e}^{\,i}_{m_i}) + \omega_{esp} \sum_{j=1}^{n\_\text{esp}_i} f_R(s^{ij}, \vec{e}^{\,i}_{m_i}) \\ & + \omega_{\text{ngh}} \sum_{j=1}^{n\_\text{ngh}_i} f_R(e^j, \vec{e}^{\,i}_{m_i}) \\ & + \omega_{\text{biton}} \sum_{\substack{p\, \in\, B(e^i_{m_i,r}) \\ I(\vec{p}) = 1}} f_A(p, \vec{e}^{\,i}_{m_i}) \\ & + \omega_{\text{add}} \sum_{\substack{j\, =\, 1 \\ j\, \neq\, i.}}^{n_e} f_A(e^j, \vec{e}^{\,i}_{m_i}).\end{aligned}$$

The first phase of the process finishes when all the *extreme* segments stop because they have reached a corresponding *extreme* segment or created a junction with an *arc* segment. The second phase starts with only one difference—the attracting contribution is reduced to the corresponding *extreme* segment l, providing it exists. The evolution equation is

$$\begin{aligned}\vec{e}^{\,i}_{m_i+1} = {}& \vec{e}^{\,i}_{m_i} + \omega_e\, f_R(e^i, \vec{e}^{\,i}_{m_i}) + \omega_{esp} \sum_{j=1}^{n\_\text{esp}_i} f_R(s^{ij}, \vec{e}^{\,i}_{m_i}) \\ & + \omega_{ngh} \sum_{j=1}^{n\_\text{ngh}_i} f_R(e^j, \vec{e}^{\,i}_{m_i}) + \omega_e f_A(e^l, \vec{e}^{\,i}_{m_i}) \\ & + \omega_{\text{add}} \sum_{\substack{j\, =\, 1 \\ j\, \neq\, i.}}^{n_e} f_A(e^j, \vec{e}^{\,i}_{m_i}).\end{aligned}$$

The gaps between edges left by edge extraction are completed by taking into account the fact that the total energy of the edges must be at a minimum. By doing this, one of the advantages of the model is achieved—we obtain regular boundaries. Some examples can be seen in the third step (red lines) of Figs. 2 and 6.

[3] If a multivariable function $V(\mathbf{x})$ is differentiable in a neighborhood of a point $\mathbf{x_0}$, then $V(\mathbf{x})$ decreases fastest if one goes from $\mathbf{x_0}$ in the direction of the negative gradient of V at $\mathbf{x_0}$.

2) Model Fitting: Once the EM has been applied, the image is completely segmented in terms of the number of fields and their approximate boundaries. But, for each boundary that has been completed, it is still necessary to analyze the ratio between the length of the segments added and the one that comes from the HED extraction. Also, isolated long edges are analyzed to decide whether they are clues to add the complete division of the field or if they represent just a groove. Moreover, it is necessary to take into account the length of the contours to avoid closing small regions over themselves. This is similar to the way that the edge completion step rejects reaching the segment that grows at the beginning of the process. These processes are done not only using the segments, but also by taking into account the image information inside the output regions. An added line in a boundary is analyzed, and even modified, by a local radiometry growing process that uses a threshold ($\mathrm{Add}_{\max}$). The boundary modified is kept when the number of nonoverlapping pixels, between the original region and the output one, is lower than $\mathrm{Add}^2_{\max}$. Otherwise, if the added pixels are lower than $\mathrm{Add}_{\max}$, the addition is admitted. Fig. 6 shows how the outputs of the *EM* phase produces edges that will be included in the final results, depending on whether these additions stay under the $\mathrm{Add}_{\max}$ parameter.

III. Experimental Results

This section includes a description of the dataset, the implementation settings, the statistical measures used for validation purposes, and the experimentation carried out to evaluate the proposed method, including a comparison with state-of-the-art methods and a discussion of the results.

A. Agricultural Field Dataset

To the best of our knowledge, there are no representative public datasets that fit our goal of segmenting agricultural fields from HRV images. For this reason, we built a complete dataset composed of 1200 HVR images and evaluated our approach on it. Moreover, it is publicly available[4,5] to serve as a benchmark for comparing the agricultural field segmentation of different algorithms. Our dataset is composed of 1200 HVR images (RGB, spatial resolution: 500 × 500 pixels), and their associated ground truth (GT) delineated by a human operator. Fig. 7(a) shows 13 original images and their corresponding GT delineated by a professional experienced in manual aerial boundary delineation.

The dataset is composed of images available on the Institut Cartogràfic i Geològic de Catalunya website[6], which are parts of 1:25.000 orthophotos. We have chosen areas with assorted agricultural field appearances and from the agricultural regions of Catalonia, such as la Plana de Lleida, Baix Camp, and Penedès (Tarragona, Spain). The flights to obtain the aerial images which form orthophotos were taken under clear weather conditions. The growing state of the crop is not important as long as the fields can be distinguished from their surroundings or are surrounded

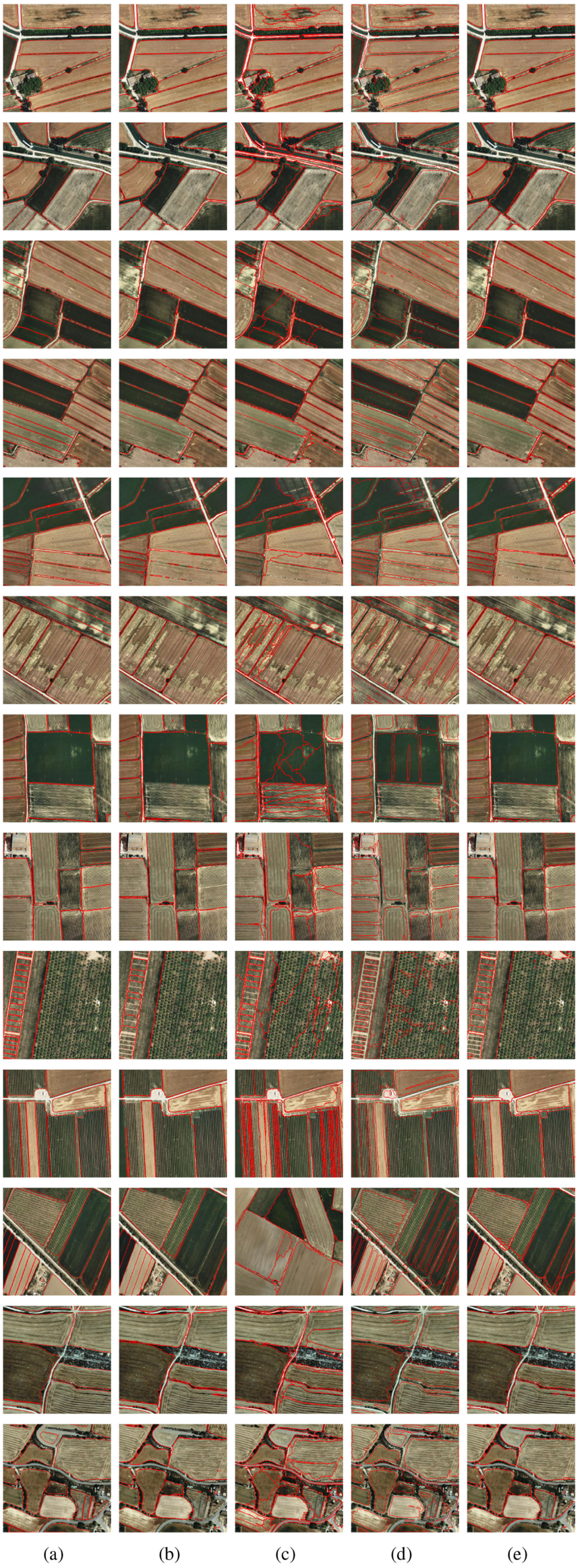

Fig. 7. (a) Input images with boundaries drawn by a human operator (GT), (b) HED results, (c) eCognition results, (d) BDCN results, and (e) DeepNEM results. These images represent a wide range of agricultural fields, in terms of radiometry and texture.

[4] Online. [Available]: http://www.aic.uniovi.es/bremeseiro/agriculturalfield-seg/

[5] Online. [Available]: https://mat-web.upc.edu/people/fernando.martinez/dataset_af-seg.html

[6] Online. [Available]: https://icgc.cat/

by linear elements, such as roads or water streams. This is one of the main advantages of mainly relying on contrast lines instead of doing it with radiometric clues. We have selected several types of crops, such as wheat, corn, hay, olive orchard, vineyard, and fruit trees.

The images contain more than 3300 agricultural fields. It is worth noting the great variability of the images, not only in terms of crops or textures, but also in size, shape, and different kind of elements acting as boundaries. Note also that the dataset contains fields with limits not completely defined, as well as others that contain some isolated elements, such as trees, bushes, or grooves.

B. Implementation Settings

The HED network delivered was trained from scratch, using an initial learning rate of 10^{-5}, which was lowered by 10 times each 1000 epochs. Note that the learning rate is important to start the training process, since the process does not converge when it is set to a value greater than 10^{-5}.

For experimentation purposes, the dataset has been split into train and test partitions with the following distribution—the training set contains 920 images, while the test set includes 280 images.

Data augmentation has proven to be a crucial technique to obtain more reliable results when dealing with DNs with a large amount of parameters, as a way to enlarge the dataset in order to train the network. In this sense, we rotated the images at 16 different angles and cropped the largest rectangle in the rotated image. Moreover, we flipped these images at each angle, leading to an augmented training set that is a factor of 32 larger than the original one. All these images were scaled to the half and to the double. To sum up, we used a total of 96×920 training images. After the training process with the augmented data, the method was tested over the 280 images, and the results were compared to the GT.

C. Performance Measures

Some quantitative metrics were used to evaluate the performance of our method. On the one hand, we consider the *Jaccard distance (JD)* between two regions A and B, which represents the number of pixels that fall into the intersection of both regions, normalized by the pixels counted by the union (also known as *intersection over union*)

$$\text{JD} = \frac{|A \cap B|}{|A \cup B|}. \tag{1}$$

For a given image, we use this equation to compute the JD between all the regions extracted from an input image and its corresponding GT regions. The *average JD* is calculated as the mean value of the JD computed for all the pairs of fields in the image.

On the other hand, the under and oversegmentation must also be considered in the problem at hand. For this purpose, we defined three different types of regions as follows:

1) type A: one extracted region that fully represents one GT field;
2) type B: two or more extracted regions that represent one GT field (oversegmentation);
3) type C: one extracted region that represents two or more GT fields (undersegmentation).

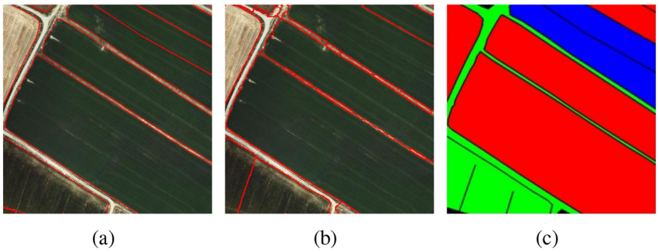

Fig. 8. (a) Input image with delineated GT, (b) results obtained with DeepNEM, and (c) results in terms of over and undersegmentation: regions type A in red, type B in green, and type C in blue.

In this case, we calculated the number of regions for each type. It should be noted that for regions of type A, the higher the number, the better; while for the regions of type B and C, the lower, the better.

Note that these distinctions among the different ways of recovering fields are necessary and highly related to human interpretation, as shown in Fig. 8. Some fields are clearly defined but, in others, the human operator may add some lines or may erase others. The automatic process needs clues and only rejects them when they do not follow the model clearly, not by interpretation. The red areas in GT are fields recovered only by a single region. On the other hand, green ones are fields clearly split into two or more areas by DeepNEM, while the opposite phenomenon is shown in blue color. For these reasons, it is necessary to compute the JD not only when the correspondence falls in type A category, but also when fields have been clearly split or merged with neighbors. In some of these last categories, human delineation could have split or merged delineations.

Additionally, we consider three region-based metrics [22], [23] commonly used in different segmentation problems.

- *Covering (CO):* It represents the level of overlapping between each pair of regions (R and R') corresponding to the GT and the output (O) images

$$CO = \frac{1}{N} \sum_{R \in GT} |R| \cdot \max_{R' \in O} \frac{|R \cap R'|}{|R \cup R'|} \tag{2}$$

where N is the number of pixels of the image.

- *Rank index (RI):* It represents the compatibility of assignments between pairs of elements in the GT and the output images

$$RI = \frac{1}{\binom{N}{2}} \sum_{i<j} [\mathbb{I}(t_i == t_j \wedge p_i == p_j) + \mathbb{I}(t_i \neq t_j \wedge p_i \neq p_j)] \tag{3}$$

where $\binom{N}{2}$ is the number of possible unique pairs among the N pixels of each image, and $\mathbb{I}$ is the identity function.

TABLE I
RESULTS OBTAINED BY DEEPNEM USING DIFFERENT VALUES FOR THE THREE PARAMETERS: $A_{\min}$, $T_{\min}$, AND $\text{Add}_{\max}$

				No. of regions				
A_{min}	T_{min}	Add_{max}	average JD	Type A	Type B	Type C	JD $\geq$ 0.9	JD $<$ 0.7
40	8	10	0.9032 ± 0.0277	**2087**	**441**	327	1745	458
		15	0.9044 ± 0.0267	2068	504	290	1766	451
		20	0.9047 ± 0.0265	2030	570	273	1790	440
		25	0.9044 ± 0.0266	2021	600	266	1797	426
		30	0.9049 ± 0.0263	2015	623	**251**	**1804**	**424**
40	6	20	0.9047 ± 0.0267	2021	581	273	1802	438
	8		0.9047 ± 0.0265	2030	570	273	1790	440
	10		$\mathbf{0.9050 \pm 0.0263}$	2019	562	287	1791	445
	12		0.9045 ± 0.0265	2003	547	300	1775	463
	14		0.9045 ± 0.0265	2003	547	300	1775	463
20	8	20	0.9049 ± 0.0264	2024	595	278	1803	436
30			0.9047 ± 0.0265	2011	584	281	1794	437
40			0.9047 ± 0.0265	2030	570	273	1790	440
50			0.9046 ± 0.0264	2035	565	272	1790	441
60			0.9047 ± 0.0263	2040	556	274	1791	443

Blanks correspond to the same value that heads the column. Bold indicates the best value for each type of measure.

- *Variation of information (VI).* It represents the distance between the GT and the output (O) images in terms of their average conditional entropy

$$VI = H(O) + H(GT) - 2 \cdot MI(O, GT) \quad (4)$$

where H and MI are the entropy and the mutual information, respectively. In this case, the lower the better.

Finally, we also consider three boundary-based metrics [23]. For this purpose, we used the edges (boundaries of the regions) to compute three standard measures commonly used in different learning tasks.

1) *Recall:* It represents the proportion of true positives correctly classified.
2) *Precision:* It represents the proportion of true positives against all the positives.
3) *F-measure:* It represents the harmonic mean of precision and recall.

Note that the three region-based metrics and the three boundary-based measures were calculated for each single image, and then the mean across images was computed.

D. Results and Discussion

1) Robustness of the Proposed DeepNEM: In terms of DeepNEM algorithm, there are three parameters that affect the capacity of the method to complete the field boundaries and to provide the final segmentation, which are as follows:

1) $A_{\min}$, the minimum isolated length element;
2) $T_{\min}$, the length of spurious segments;
3) $\text{Add}_{\max}$, the number of pixels allowed to be added for each boundary.

In order to analyze their impact on the performance results, we tested different values for them. Table I shows these values and the measures obtained for each parameter configuration. As can be observed, DeepNEM is a robust method that provides very competitive and stable results regardless of small changes in the parameters. Depending on the purpose of the segmentation, priority may be given to obtaining more type A regions or having more regions with a distance of JD above 0.9. Taking into account the problem at hand, the best tradeoff for all the metrics evaluated is achieved when the parameter configuration is: $A_{\min} = 40$, $T_{\min} = 8$, and $\text{Add}_{\max} = 20$.

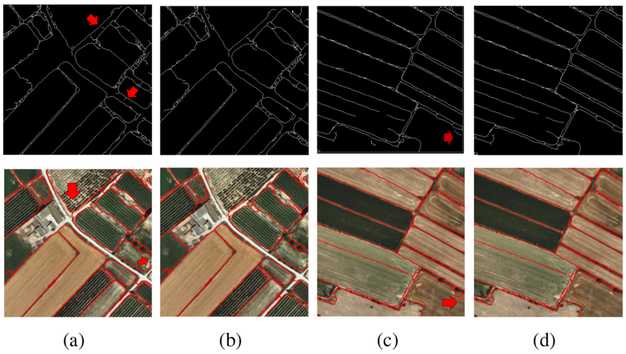

Fig. 9. Output edges obtained after the *edge map construction* (top), and their corresponding regions (bottom). (a), (b) Results obtained with parameter $A_{\min} = 40$ and different $T_{\min}$ values. (a) $T_{\min} = 6$ allows to keep more detailed segments and to complete the two boundaries pointed out by the arrows (JD = 0.8950, type A = 15). (b) $T_{\min} = 14$ provides less detailed segments and so these two boundaries are lost (JD = 0.8937, type A = 14). (c), (d) Results obtained with parameter $T_{\min} = 8$ and different $A_{\min}$ values. (c) $A_{\min} = 20$ generates more division inside fields (JD = 0.9053, type A = 12). (d) $A_{\min} = 60$ deletes pixel areas (JD = 0.9051, Ttype A = 13). Since the minimization process acts globally, some details that are kept depending on the parameters will affect other neighboring fields, as it is pointed out with the red arrows in the final results (a). Some parts of fields are completed meanwhile these regions will not be recovered without these clues (b).

Finally, Figs. 9 and 10 show the impact of these parameters by means of some representative examples. First, Fig. 9 shows the effects of the parameters $A_{\min}$ and $T_{\min}$. As can be observed, the lower they are, the more details are kept and the more likely the results are to present oversegmentation.

TABLE II
PERFORMANCE MEASURES OBTAINED WHEN APPLYING THE THREE DIFFERENT METHODS TO THE 280 TEST IMAGES

	HED	eCognition	BDCN	DeepNEM
average JD	0.8983 ± 0.0367	0.8870 ± 0.0324	0.7916 ± 0.0892	$\mathbf{0.9047 \pm 0.0265}$
No. of type A regions (1 to 1)	1759	1301	1305	**2030**
No. of type B regions (over-segmentation)	**63**	795	232	570
No. of type C regions (under-segmentation)	756	478	**112**	273
Covering	0.590	0.609	0.589	**0.782**
Rank index	0.880	0.849	**0.888**	0.874
Variation of information	0.442	0.530	**0.387**	0.474
Recall	0.571	0.654	0.490	**0.679**
Precision	0.543	0.563	0.491	**0.581**
F-measure	0.557	0.604	0.491	**0.626**

The average JD is in terms of mean ± standard deviation, calculated across all the test images. Bold indicates the best value for each type of measure.

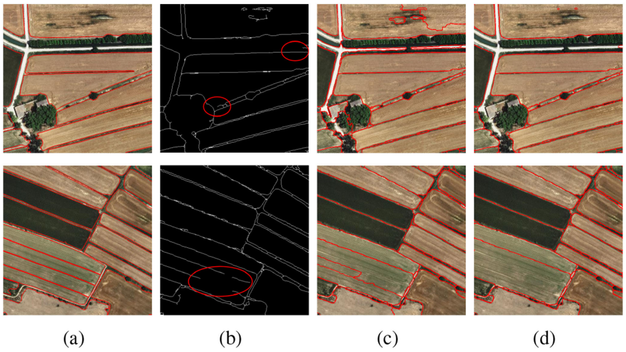

Fig. 10. Examples of oversegmentation (top) and undersegmentation (bottom). (a) Input images with their delineated GT, (b) HED results, (c) DeepNEM results after EM, and (d) final DeepNEM results. (Top) The JD is similar in (d) and (e), 0.8860 and 0.8853, respectively; whereas the number of type A regions increases from 7 to 8. (Bottom) The JD improves from (d) to (e), 0.8947 and 0.9051, respectively; while the number of type C regions increases from 0 to 3, because the three parcels highlighted (red circle) are recovered as only one.

Regardless of the established parameters, it is difficult to recover human interpretation. DeepNEM can reinforce this division or just ignore it, because there are not enough clues for it. This fact is recovered by fitting the results to a model, as it is shown in Fig. 10 (top). The opposite phenomenon is also shown in Fig. 10 (bottom): the division, due to minimization process, delivers a segmentation coincident with the GT in terms of region number, but their boundaries are different since the process is driven by radiometric information. The final model constraint will erase this division because the number of boundary pixels added is greater than a $\mathrm{Add}_{\max}$.

2) Comparison to the State-of-the-Art: We compared DeepNEM with the following three state-of-the-art methods for aerial segmentation:

1) HED [12], in order to reveal the improvement achieved by adding the proposed EM framework;
2) eCognition [1], the commercial software most widely used in remote sensing field;
3) BDCN [16], one of the most recent frameworks for edge detection and segmentation.

We applied DeepNEM as well as these three approaches to the 280 test images of the dataset described in Section III-A.

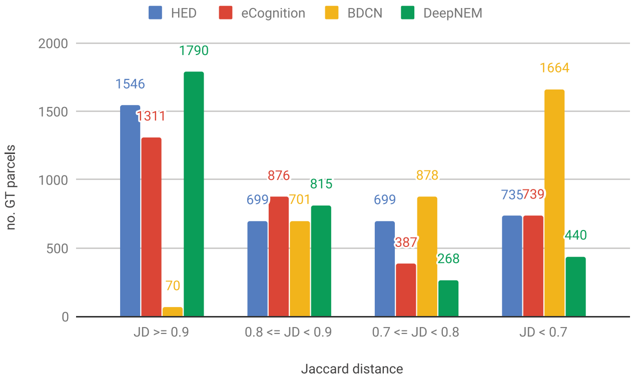

Fig. 11. Performance of the different methods in terms of the number of GT fields segmented, for different values of JD.

Regarding the parameter settings of DeepNEM, we used the most competitive ones according to the experimentation presented in Section III-D1: $A_{\min} = 40$, $T_{\min} = 8$ and $\mathrm{Add}_{\max} = 20$. With respect to the BDCN network, it was trained from scratch using an initial learning rate of 10^{-6}, which was reduced by 10 times each 10 000 epochs. Other configuration parameters include a momentum of 0.9, and weight decay of 2^{-4}.

Table II includes the results of the four methods considered (HED, eCognition, BDCN, and DeepNEM) in terms of the average JD calculated over the 280 test images. As can be seen, the worst results are obtained with eCognition; while the best results are achieved when using DeepNEM, demonstrating the adequacy of the EM process applied to the edges provided by HED. The results achieved by DeepNEM are not only better on average (mean), but also have a lower standard deviation.

Fig. 11 shows the results obtained with the four different methods in four intervals of JD—from greater than or equal to 0.9, to lower than 0.7. As can be observed, DeepNEM obtains the best results by detecting 1790 regions with a JD $>= 0.9$, which represents 54.03% of all the fields detected. This method is followed by HED, demonstrating once again the adequacy of using DNs in the problem at hand, and providing a better performance than the commercial software eCognition and the novel BDCN, which has a lower performance in this case. Analyzing the fields with a low JD, it should be highlighted

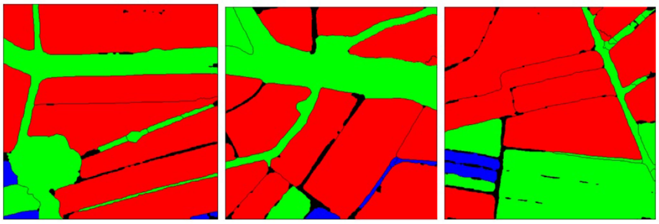

Fig. 12. Graphical representation of *JD* for three of the five images shown in Fig. 7: JD ≥ 0.9 (green), $0.8 \leq$ JD < 0.9 (red), and $0.7 \leq$ JD < 0.8 (blue). Note that all regions are types A or B.

that only 13.28% of the regions detected by DeepNEM have a JD < 0.7.

Table II also shows the results in terms of the different types of regions (A, B, and C). Regarding the fields successfully recovered (type A), the numbers obtained when applying eCognition are noticeably lower than when using DNs, showing that eCognition tends to produce over or undersegmentation. HED achieves the best results in terms of oversegmentation (type B), with only 63 regions showing that it tends to produce undersegmentation. On the other hand, BDCN provides the best results in terms of undersegmentation (type C), with only 112 regions. In fact, BDCN shows the best balance between types B and C regions. However, the number of fields 1 to 1 is quite low, similar to the one achieved by eCognition. The most competitive results in terms of 1 to 1 regions (type A) are achieved by DeepNEM, which also provides a good tradeoff with respect to the number of over and undersegmented fields (types B and C, respectively). In order to illustrate the different types of fields associated to aerial image segmentation, Fig. 12 uses different colors to show how DeepNEM works on three sample images. As can be seen, almost all the regions are extracted one to one with a JD ≥ 0.9.

Regarding the region- and boundary-based metrics, they are also reported in Table II. As can be observed, DeepNEM outperforms the other three methods in four out of the six measures (covering, precision, recall, and f-measure), followed by eCognition. With respect to the other two metrics (rank index and variation of information), the best results are provided by BDCN. However, it is worth noting that there are no significant differences in terms of the rank index, with very similar results achieved regardless the method considered.

Fig. 7 shows 13 images for a qualitative comparison. Fig. 7(a) shows the original images with their corresponding GT stacked, Fig. 7(b) shows the results obtained with HED (using the fusion layer as output), Fig. 7(c) shows the results obtained with eCognition, Fig. 7(d) shows the results obtained with BDCN, and Fig. 7(e) shows the results obtained with DeepNEM. As can be observed, DeepNEM keeps the boundaries more clearly delineated. On the other hand, there are more regions oversegmented with eCognition than with DeepNEM. This oversegmentation is due to radiometric variability inside each field, which is relevant enough to be recovered by a software that relies on radiometry. On the other hand, DeepNEM relies more on linear elements, since it has been trained to find these clues, and introduced constraints to keep these elements in the EM process. Note that eCognition is used in productive environments as a classification-segmentation software. So, it is very important to evaluate results in terms of overlapping areas, as well as to analyze the way in which areas are recovered—under and oversegmentation. This fact determines the amount of manual edition necessary to obtain a final product. Note that DeepNEM reduces by a quarter the necessary edition obtaining much less under and oversegmented images.

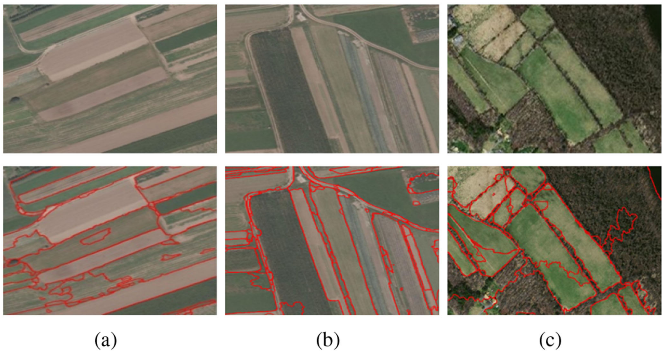

Fig. 13. Original images (top) and DeepNEM results (bottom). (a) and (b) Original images from the AIRS dataset. (c) Original image from the Massachusetts dataset.

3) Aerial Datasets: As far as the authors know, there is no public dataset for aerial segmentation. However, we have found two aerial datasets for object detection (instead of field segmentation)—AIRS [24] and Massachusetts Buildings [25]. Despite this, it is possible to identify some agricultural regions among their images. We used them in order to check how robust is our algorithm run on different datasets.

From these datasets, we selected the images that contain fields, and ran our DeepNEM. The results are shown in Fig. 13. Due to the fact that their resolution is smaller than that associated with our training dataset, DeepNEM has to rely more on constraints associated with the model than on the edges extracted by the HED. For this reason, the results are slightly oversegmented in fields that are highly textured. Anyway, the improvement of the results over these images makes necessary GT associated to these agricultural fields, and train the DN with them.

IV. Conclusion

We present a joint venture between a DN and an EM model-guided radiometric method that improves the benefits of each component. The two-step process we proposed, represented by DeepNEM, has been trained and tested over a new public aerial dataset of 1200 images. The contours delivered by our DeepNEM are really close to the GT both in area and shape. Furthermore, it is possible to take advantage of the by-products in order to trigger other semiautomatic segmentation processes, as we have demonstrated in the validation section. The two step process can improve as other networks deliver better edges. DeepNEM has been tested over a variety of natural areas and compared with other region extraction algorithms. This has demonstrated that DeepNEM eliminates the need for human interaction and obtains smoother and more reliable results. When the image is a continuum of regions, DeepNEM will pull them apart, whether or not there is any evidence of border reliable enough. Moreover, if inside the fields, there are some trees or bushes, which are not large enough to become an isolated entity, the process will not consider them.

Nowadays, the earth is continuously monitored with different types of sensors at different resolutions. As for future lines of research, it will be not only important, but also effective to train the system with images at different resolutions and see how a single network manages this variety. Furthermore, we plan to investigate how to diversify the model to cope with this multiresolution. Further research also includes the evaluation of a network of networks to solve this challenge.

Acknowledgment

The authors would like to thank the support of NVIDIA Corporation with the donation of the Titan Xp GPUs.

References

[1] Y. Zhang and T. Maxwell, "A fuzzy logic approach to supervised segmentation for object-oriented classification," in *Proc. ASPRS Annu. Conf.*, 2006, pp. 1–5.

[2] S. Chen and D. Zhang, "Robust image segmentation using FCM with spatial constraints based on new kernel-induced distance measure," *IEEE Trans. Syst., Man, Cybern., B: Cybern.*, vol. 34, no. 4, pp. 1907–1916, Aug. 2004.

[3] M. D. Hossain and D. Chen, "Segmentation for object-based image analysis (OBIA): A review of algorithms and challenges from remote sensing perspective," *ISPRS J. Photogrammetry Remote Sens.*, vol. 150, pp. 115–134, 2019.

[4] S. Bhardwaj and A. Mittal, "A survey on various edge detector techniques," *Proc. Technol.*, vol. 4, pp. 220–226, 2012.

[5] M. M. Alemu, "Automated farm field delineation and crop row detection from satellite images," Ph.D. dissertation, Fac. Geo Informat. Sci. Earth Observ., Univ. Twente, Enschede, Netherlands, 2016.

[6] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A line segment detector," *Image Process. Line*, vol. 2, pp. 35–55, 2012.

[7] Y. Boykov and G. Funka-Lea, "Graph cuts and efficient nd image segmentation," *Int. J. Comput. Vision*, vol. 70, no. 2, pp. 109–131, 2006.

[8] L. Grady, "Random walks for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1768–1783, Nov. 2006.

[9] A. K. Sinop and L. Grady, "A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm," in *Proc. IEEE Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[11] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1395–1403.

[12] S. Xie and Z. Tu, "Holistically-nested edge detection," *Int. J. Comput. Vision*, vol. 125, no. 1/3, 2017, Art. no. 3.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, *arXiv:1409.1556*.

[14] Y. Liu and M. S. Lew, "Learning relaxed deep supervision for better edge detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 231–240.

[15] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3000–3009.

[16] J. He, S. Zhang, M. Yang, Y. Shan, and T. Huang, "Bi-directional cascade network for perceptual edge detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 3828–3837.

[17] A. Troya-Galvis, P. Gançarski, and L. Berti-Équille, "Remote sensing image analysis by aggregation of segmentation-classification collaborative agents," *Pattern Recognit.*, vol. 73, pp. 259–274, 2018.

[18] O. Csillik, "Fast segmentation and classification of very high resolution remote sensing data using SLIC superpixels," *Remote Sens.*, vol. 9, no. 3, 2017, Art. no. 243.

[19] H. Gu *et al.*, "An efficient parallel multi-scale segmentation method for remote sensing imagery," *Remote Sens.*, vol. 10, no. 4, 2018, Art. no. 590.

[20] L. Mou, Y. Hua, and X. X. Zhu, "A relation-augmented fully convolutional network for semantic segmentation in aerial scenes," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 12 416–12 425.

[21] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Artif. Intell. Statist.*, 2015, pp. 562–570.

[22] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[23] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: A new dataset, experiments, and results," *IEEE J. Biomed. Health Informat.*, vol. 21, no. 3, pp. 588–598, May 2017.

[24] Q. Chen, L. Wang, Y. Wu, G. Wu, Z. Guo, and S. L. Waslander, "Aerial imagery for roof segmentation: A large-scale dataset towards automatic mapping of buildings," *ISPRS J. Photogrammetry Remote Sens.*, vol. 147, pp. 42–55, 2019.

[25] V. Mnih, "Machine learning for Aerial image labeling," Ph.D. dissertation, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 2013.

Margarita Torre received the bachelor's degree in mathematics from the Universitat de Barcelona, Barcelona, Spain, in 1987, and the bachelor's degree in computer sciences from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1999. She is currently working toward the Ph.D. dregree with the Computer Vision Center of the Universitat Autònoma de Barcelona, Bellaterra, Spain.

She has developed several applications and productive systems for cartographic purposes. For 20 years she worked with the Institut Cartogràfic de Catalunya, Barcelona, Spain, where, from 1997 to 2007, she was the Head of the Digital Photogrammetric Development Department. Since 2007, she has been working with the Research and Innovation area of the Department of Territory and Sustainability, Generalitat de Catalunya, Barcelona, Spain. Her research interests include techniques from computer vision to extract knowledge from images, deformable models, and its simplification.

Beatriz Remeseiro received the B.S., M.S., and the Ph.D. (Cum Laude with International Honors) degrees in computer science from the University of A Coruña, A Coruña, Spain, in 2008, 2010, and 2014, respectively.

After two Postdoctoral Fellowships from 2015 to 2017, with the INESC TEC, Porto, Portugal, and the University of Barcelona, Barcelona, Spain, she is currently an Assistant Professor with the University of Oviedo, Gijón, Spain. She has coauthored five book chapters and more than 50 research papers in international journals and conferences. She has a coorganized several special sessions at different international conferences, and served in program and scientific committees. Her main research interests include computer vision and deep learning, mainly applied to real-world problems.

Dr. Remeseiro's Ph.D. thesis was honored with the "2nd Award to the Best Ph.D. Thesis 2014" from the Spanish Association for Artificial Intelligence. She was the recipient of the "Gradiant Award to the best Ph.D. Thesis applied to ICT 2016" from the Galician Official College and Association of Telecommunications Engineering.

Petia Radeva (Fellow, IEEE) is a Full Professor with the Universitat de Barcelona (UB), Barcelona, Spain, PI with the Consolidated Research Group Computer Vision and Machine Learning, UB and a Senior Researcher with Computer Vision Center. She was PI with UB in four European, three international, and more than 20 national projects devoted to applying computer vision and machine learning for real problems, such as food intake monitoring (e.g., for patients with kidney transplants and for older people).

Prof. Radeva is a REA-FET-OPEN Vice-Chair since 2015 onward, and an International Mentor with the Wild Cards EIT program, since 2017. She is an Associate Editor of *Pattern Recognition* journal (Q1) and *International Journal of Visual Communication and Image Representation* (Q2). She has been awarded IAPR Fellow since 2015, ICREA Academia assigned to the 30 best scientists in Catalonia for her scientific merits since 2014, was the recipient of several international awards ("Aurora Pons Porrata" of CIARP, Prize "Antonio Caparrós" for the best technology transfer of UB, etc.). She has supervised 16 Ph.D. students and authored or coauthored more than 90 SCI journal publications and 250 international chapters and proceedings, her Google scholar h-index is 43 with more than 7000 cites and WOS h-index: 79.

Fernando Martínez received the Bachelor's degree in physics from the Universitat de Barcelona, Barcelona, Spain, in 1989, and the Ph.D. degree in mathematics from the Universitat Politècnica de Catalunya, Barcelona, Spain, in 1997.

He has worked in partial differential equations applied to elasticity and later to computer vision. Currently, he is an Associate Professor with the Department of Mathematics, Universitat Politècnica de Catalunya.