Journal of Electrical and Computer Engineering Innovations
(JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

Automatic Cadastral Boundary Detection of Very High Resolution Images Using Mask R-CNN

N. Rahimpour Anaraki, A. Azadbakht, M. Tahmasbi *, H. Farahani, S. R. Kheradpishe, A. Javaheri

Department of computer and data sciences, Faculty of mathematical sciences, Shahid Beheshti University, Tehran, Iran.

Article Info

Article History:
Received 05 March 2024
Reviewed 15 April 2024
Revised 03 July 2024
Accepted 15 July 2024

Keywords:
Remote sensing
Mask R-CNN
Cadastral mapping
Instance segmentation

*Corresponding Author's Email Address:
m_tahmasbi@sbu.ac.ir

Abstract

Background and Objectives: Cadastral boundary detection deals with locating the boundary of the ownership and use of land. Recently, there has been high demand for accelerating and improving the automatic detection of cadastral mapping. As this problem is in its starting point, there are few researches using deep learning algorithms.

Methods: In this paper, we develop an algorithm with a Mask R-CNN core followed with geometric post-processing methods that improve the quality of the output. Many researches use classification or semantic segmentation but our algorithm employs instance segmentation. Our algorithm includes two parts, each of which consists of a few phases. In the first part, we use Mask R-CNN with the backbone of a pre-trained ResNet-50 on the ImageNet dataset. In the second part, we apply three geometric post-processing methods to the output of the first part to get better overall output. Here, we also use computational geometry to introduce a new method for simplifying lines which we call pocket-based simplification algorithm.

Results: We used 3 google map images with sizes 4963 × 2819, 3999 × 3999, and 5520 × 3776 pixels. And divide them to overlapping and non-overlapping 400×400 patches used for training the algorithm. Then we tested it on a google map image from Famenin region in Iran. To evaluate the performance of our algorithm, we use popular metrics Recall, Precision, and F-score. The highest Recall is 95%, which also maintains a high precision of 72%. This results in an F-score of 82%.

Conclusion: The idea of semantic segmentation to derive boundary of regions, is new. We used Mask R-CNN as the core of our algorithm, that is known as a very suitable tools for semantic segmentation. Our algorithm performs geometric post-process improves the f-score by almost 10 percent. The scores for a region in Iran containing many small farms is very good.

This work is distributed under the CC BY license (http://creativecommons.org/licenses/by/4.0/)

Introduction

One of the bases of land administration systems is recording the ownership and physical location of real properties, which are called cadastres [1]. Recently, cadastral mapping has received considerable attention. An effective cadastral system formalizes private property rights, which is very important to promote agricultural productivity, support national development, and secure an effective land market [2].

However, estimates suggest that about 75% of the world's population doesn't have access to a formal system to register and safeguard their land rights. Establishing a complete land cadaster and keeping it up-to-date is a contemporary challenge for many developing and developed countries [3], [4]. This lack of recorded

land rights increases insecure land tenure and fosters existence-threatening conflicts, particularly in developing countries. Recording land rights spatially, i.e., cadastral mapping by traditional field surveying approaches, is considered the most expensive, time-consuming, and labor-intensive part of a land administration system. Therefore, in order to speed up the process, we need innovative tools [5], [6].

Earth observation satellites provide very high-resolution (VHR) images. Unmanned aerial vehicle (UAV) images are also available in different areas. According to the Union of Concerned Scientists (UCS), there were 971 EO satellites in orbit on the 30th of April 2021. For context, when they did a similar report at the end of April 2018, there were only 684 satellites, so there has been a 41.95% increase over the three years [7].

Since the availability of VHR images, remote sensing has been used for mapping cadastral boundaries instead of field surveying and is advocated by fit-for-purpose (FFP) land administration [3]. In these images, visible cadastral boundaries are often marked by physical objects such as rivers, roads, water drainages, building walls, clusters of stones, fences, strips of uncultivated land, ditches, etc. [1]. These boundaries are apparent in remotely sensed images and can be automatically identified using image processing algorithms [6].

Recent studies have highlighted the effectiveness of deep learning techniques, such as Convolutional Neural Networks (CNNs), in extracting high-level representations crucial for detection and classification tasks, opening up new opportunities in cadastral boundary detection [8,9].

In deep learning, two primary approaches are commonly employed to train CNNs: starting from scratch or utilizing transfer learning [10]. In this study, we opt for the Mask R-CNN model with a pre-trained ResNet-50 backbone from the ImageNet dataset for instance segmentation to identify cadastral boundaries. This approach, initially used by Mayer et al. [11], focuses on delineating the boundaries of individual fields. Here we focus on finding the boundary of each individual farm and leveraging transfer learning for pre-trained ResNet-50 was used. Also, we introduce a new geometric method for simplifying lines (extracted boundary) which we call the Pocket-based simplification algorithm that achieves better performance than the commonly used Douglas-Peucker algorithm [12].

The Core of our algorithm is a Mask R-CNN with the backbone of a pre-trained ResNet-50 on the ImageNet dataset to produce the initial output, which is a probability map of fields or non-field pixels. Then, Otsu's binary method with a certain threshold is applied to the output to identify which pixels represent fields. Finally, Canny edge detection algorithm delineates the boundaries of these fields. In the second part, to enhance the results, we eliminate fields that have very small area. Then, we remove polygons inside other polygons, and finally, the extracted lines (boundaries) are simplified using a new geometric simplification algorithm. The third part involves accuracy assessment where we add a buffer to the ground-truth boundaries and calculate Recall, Precision, and F-score in various scenarios.

This paper is organized as follows: in section Literature Review we review recent results in this field. In the section Method, we first describe our algorithm in detail and evaluation methods. Details on the training data are provided in the section Experimental Study. Section Evaluation following by section results and discussion, focuses on assessing the model and comparing our proposed simplification algorithm with the state-of-the-art. Finally, in the last section, we present our final conclusion and open problems.

Literature Review

In this section, we review some studies in this field and discuss various image processing methods to address problems.

Some methods for cadastral mapping are based on image segmentation and edge detection. In Drăguţ et al. [13], they introduce a new automated approach called Multi-Resolution Segmentation (MRS) for parameterizing multi-scale image segmentation of multiple layers. This approach relies on the potential of local variance to detect scale transitions in geospatial data. Classical edge detection aims to identify sharp changes in image brightness through local measurements, including first-order (e.g., Prewitt or Sobel) and second-order (e.g., Gaussian or Laplacian) derivative-based detection [14].

In a study by Crommelinck et al. [15] they aimed to apply computer vision techniques to analyze remotely sensed UAV images for UAV-based cadastral mapping. Their approach involved a three-step process for identifying cadastral boundaries. The first step is image preprocessing, where the UAV orthoimage underwent resampling to lower resolutions and was divided into patches. Next, in boundary delineation, they utilized the Globalized Probability of Boundary (gPb) contour detection method, the state-of-the-art computer vision method, on each patch. This process generated contour maps with probabilities assigned to contours per pixel. Lastly, in image post-processing, all patches from the same image were combined to create a unified contour map and a binary boundary map, which was then converted into vector format. This systematic procedure allowed for the representation of cadastral boundaries for mapping applications.

Numerous studies have employed Convolutional Neural Network (CNN) tools for cadastral mapping, with Crommelinck et al. [16] presenting a three-step workflow. The first step involves image segmentation to extract

visible object outlines, followed by boundary classification to predict the likelihood of boundaries for the extracted segment lines. The final step is interactive delineation, connecting these lines based on the predicted boundary likelihood.

For image segmentation, Multiresolution Combinatorial Grouping (MCG) generates closed contours that show the outlines of objects. Then, a boundary classification is applied to the resulting post-processed MCG lines, using two machine learning approaches: Random Forest (RF) and pre-trained VGG19 through transfer learning, to achieve boundary likelihood per line. Interactive delineation facilitates the creation of final cadastral boundaries through various functions as a plugin in QGIS. All training images used in their study are obtained from UAV images.

In Fetai et al. [17], UAV images are utilized in their workflow, which comprises three main steps: Image pre-processing, boundary detection and extraction, and data post-processing. The initial step involves resampling the UAV orthoimage, followed by applying the ENVI feature extraction module [18], [19] to each down-sampled UAV orthoimage. In the final step, extracted objects are filtered and simplified.

Xia et al. [6] utilize deep Fully Convolutional Networks (FCNs) to detect cadastral boundaries using UAV images captured over urban and semi-urban areas. They approach boundary detection as a supervised pixel-wise image classification task to differentiate between boundary and non-boundary pixels. The network employed in their research is a modified version of the FCN with dilated kernel (FCN-DK) detailed in [20]. Other studies in this field can be found in [21]-[23].

Our solution is based on instance segmentation, which has become one of the relatively important, complex, and challenging areas in computer vision research. Hafiz et al. [24] review some advances in instance segmentation:

Girshick et al. [25] were among the first to explore CNNs for instance segmentation [26]. They developed the R-CNN technique, which integrated AlexNet [27] along with a region proposal using the selective search technique [28]. However, it has some drawbacks, such as having difficult and slow training. Therefore, Girshick [29] introduced Fast R-CNN, which addressed some of the issues of R-CNN and improved its object detection ability.

In Zagoruyko et al. [30], MultiPath Network was introduced by applying three modifications to the standard Fast R-CNN model. Initially, skip connections have been integrated to grant the object detector access to features from various network layers. Subsequently, a foveal component has been used to leverage the context of objects across varying resolutions. Lastly, an integral nature loss function has been included.

Although Fast R-CNN improved detection speed significantly, it still depended on external region proposals, which posed a computational bottleneck. To address this issue, Ren et al. [31] introduced the Faster R-CNN model with a Region Proposal Network (RPN) for efficient and accurate region proposal generation. This model utilized the same backbone network and extracted features from the last shared convolutional layer for both RPN-based region proposal and Fast R-CNN region classification.

Finally, He et al. [32] introduced Mask R-CNN as a straightforward and adaptable model for instance segmentation. This model efficiently achieves instance segmentation by combining object detection with the concurrent creation of precise masks. Mask R-CNN builds upon the foundation laid by Faster R-CNN. Typically, Faster R-CNN includes a branch dedicated to recognizing object bounding boxes. Mask R-CNN enhances this framework by introducing a parallel object mask prediction branch, thereby improving the overall performance of the model.

Method

In this section, we present our algorithm that consists of two main parts: A deep convolutional network that detects boundaries, and a geometric post-process that simplifies the boundary and cleans up the map.

A. R-CNN

Here, we propose an effective pipeline for detecting the boundaries of two fields in satellite images. The core module in this pipeline is a Mask R-CNN model. This model is responsible for detecting each field in small input images of size 400*400. The R-CNN is a two-stage detection algorithm. The first stage identifies a subset of regions in an image that might contain an object, i.e., a field. The second stage classifies the object in each region and returns a probability map of the presence of an object, i.e., a field over pixels.

The input images were raw images without additional labels or external data. In order to find the borders of fields in a supervised manner, we manually generated labels for some of the input images. Using the LabelMe [33] library, we carefully generated a mask for each input image, coloring each field area from 1 to 256 such that neighboring fields have different colors.

To train an instance segmentation model, we set the Mask R-CNN model with the backbone of a pre-trained ResNet-50 [34] on the Imagenet dataset [35]. ResNet-50 is selected as the backbone for its ability to convert input images into feature vectors of size 4096. These feature vectors play a crucial role in predicting the bounding box, class, and mask of each identified object. Without leveraging ResNet-50 as the backbone, the model would require a much larger amount of data for training purposes. Deeper ResNet networks require more computational power, and since we are working with super-large images, it is suitable choice. Finetuning a pre-

trained model would give better results in our experiments. Choosing the input size of this model is a tricky challenge. Small input images would cause the detected object to be more fragmented, while large input images would cause the model to ignore small features and consider small fields as noise. We chose an input size of 400*400 so that the models would detect enough small features and fields, and the output fields would not be too fragmented, while keeping the computational cost of training the model reasonable.

A dataset of 400*400 patches was extracted from the input images, and R-CNN models were trained on them. The model tries to detect each field as an individual object, and for each object, it returns a probability mask stating whether a pixel is in a field or not. In the challenge, we do not need each field alone, but we want some sense of the borders. So, if we could only identify the interior area of each field, we can extract the borders. To simplify the overall solution, we apply max pooling to all the detected object masks.

Our solution for this challenge consists of five steps: patchify, R-CNN model, unpatchify, edge and border detector, and border refiner. During inference time, we slide a window of size 400*400 with steps of 200 pixels on the input image. This way, the patches overlap. This step is called patchify. Each patch goes into the RCNN model, and the resulting masks are aggregated together to create the final large output. This step is called unpatchify. On the overlapping parts of patches, we apply different methods to aggregate the probabilities: max pooling, averaging, summation, and harmonic mean.

At the end of the unpatchify step, we obtain a large mask of the input model where values close to 1 are more likely to be fields. In order to extract borders from masks, we apply Otsu's binarizing method [36] to binarize probability masks adaptively; then the canny edge detection algorithm [37] applies to the output, and the resulting image contains only edges and borders of the fields.

Mask R-CNN is an advanced model for object detection and instance segmentation, building upon the foundations of Faster R-CNN. Faster R-CNN itself is designed to accurately identify and localize objects within an image by predicting both their bounding boxes and class scores. Mask R-CNN enhances this by introducing an additional branch that predicts segmentation masks for each identified object instance, providing a more detailed analysis of the image content. This capability is particularly useful in scenarios where precise object outlines are required, beyond mere detection.

In this study, we introduce a refined instance segmentation framework based on the integration of the Mask Region-based Convolutional Neural Network (Mask R-CNN) with a ResNet-50 architecture, pre-trained on the Comprehensive Object Detection, Localization, and Segmentation (COCO) dataset. This framework is engineered to perform dual functions: object detection and instance segmentation, enabling precise pixel-wise delineation of objects within an image. Utilizing the concept of transfer learning, the pre-trained model serves as a foundational backbone, facilitating accelerated training convergence and reducing the necessity for extensive domain-specific data.

The core adaptation of the model revolves around the customization of the Region of Interest heads, specifically the box predictor and the mask predictor, to cater to a user-defined number of object classes. The adjustment process begins with the box predictor, where the input features for the classifier are derived from the pre-existing model structure.

Subsequently, the original Fast R-CNN predictor is substituted with a novel Fast R-CNN Predictor, recalibrated to project the model's output to the specified number of classes. This alteration enables the adapted model to classify objects into a bespoke set of categories, diverging from the standard categorization learned during its initial training on the COCO dataset.

Parallel to the modifications in the box predictor, the mask predictor undergoes a similar transformation. The original mask predictor is replaced with a newly instantiated Mask R-CNN Predictor. This adjustment entails the configuration of input features alongside the introduction of an intermediate hidden layer with 256 nodes, culminating in the ability to generate class-specific segmentation masks.

Data augmentation is a critical technique in the field of machine learning and computer vision, particularly beneficial for enhancing the robustness and generalization capabilities of deep learning models. By artificially expanding the training dataset through various transformations, data augmentation introduces a diversity of perspectives, angles, and environmental conditions, simulating a more comprehensive range of real-world scenarios.

This process significantly mitigates the risk of overfitting, as the model is trained on a broader spectrum of data instances, improving its ability to generalize to unseen data. Moreover, data augmentation is applied. For this purpose, mirrored versions of the training images are added. Doing so ensures that the model is not biased towards the original orientation of objects within the images, fostering an ability to accurately detect and segment objects regardless of their horizontal alignment.

The model, leveraging a custom adaptation of the Mask R-CNN architecture with a ResNet-50 backbone, is trained using a Stochastic Gradient Descent (SGD) optimizer. This choice is motivated by SGD's proven efficacy in handling noise and its capacity for

generalization in large-scale data scenarios. The learning rate is set at 0.005, a value that balances the trade-off between training speed and the risk of overshooting minima in the loss landscape. To enhance the optimization process, a momentum of 0.9 is set, facilitating the acceleration of the optimizer in relevant directions and improving convergence speed.

Additionally, a weight decay of 0.0005 is applied as a regularization measure to prevent overfitting by penalizing large weights, ensuring the model's generalizability to unseen data.

The learning rate is reduced by a factor of 10 every 3 epochs. This approach, known as step decay, is instrumental in fine-tuning the model's performance by gradually decreasing the learning rate, allowing for more refined adjustments to the model weights as training progresses.

The training process spans 50 epochs, a duration determined to strike an optimal balance between achieving sufficient model convergence and avoiding excessive computational expenditure. The choice of 50 epochs is also influenced by empirical observations of model performance over time, ensuring that the model benefits from prolonged exposure to the training data without succumbing to overfitting.

B. Geometric Post-Process

Having a clean output as a binary PNG image, we need to transform it into a shapefile that can be used in Geographic Information Systems (GIS) like QGIS, ArcGIS, etc. Therefore, we extract polygons from the output image and after vectorizing it, a shapefile is created. The resulting polygons need further post-processing. First, we need to define some concepts, then we investigate post-process steps.

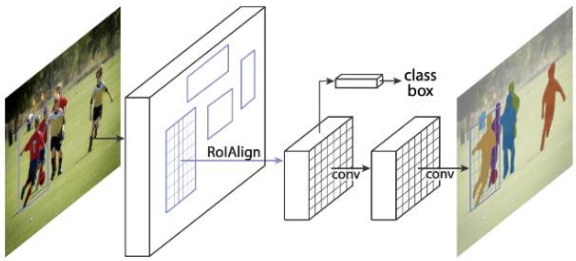

Fig. 1: Mask RCNN model.

A subset S of the plane is called convex if and only if for any pair of points $p, q \in S$ the line segment $\overline{pq}$ is completely contained in S. The *convex hull* of a set S (denoted by $CH(P)$) is the smallest convex set that contains S. To be more precise, it is the intersection of all convex sets that contain S [38].

The *pockets* of a simple polygon are the polygonal areas outside the polygon, but inside its convex hull [38]. In Fig. 2, three white polygons are pockets.

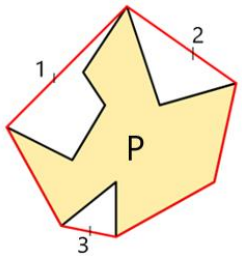

Fig. 2: Polygon P and its three pockets in white.

Our post-process performs three steps:

1. Deleting small polygons: Polygons with an area less than a fixed number with respect to Ground Sample Distance (GSD) of an image are removed because they have an area less than a real farm in that region.

2. Deleting polygons inside other polygons: We want to make sure no polygon contains any smaller polygon inside.

3. Simplifying over-segmentation: Over-segmentation can cause serrated edges in polygons. There are different simplification methods such as the Douglas-Peucker algorithm. We develop a new algorithm called the pocket-based simplification algorithm. Here is how this algorithm works:

Given a polygon P, we compute all of its pockets, $P_1 \dots P_k$. Now for each pocket, we calculate two distances:

1. $dist$: The length of the edge that belongs to the pocket but not the polygon P. We call this edge *probable edge* (numerized marked edges in Fig. 1).
2. d: The summation on the length of all edges belongs to pocket except the probable edge.

Then we choose certain threshold (denoted by t) and if $d < t \times dist$, all edges used for calculating d are added as edges of polygon P'. Else, only the probable edge is added as an edge of polygon P'. At the end of the process, polygon P' which is the simplified version of polygon P is added to the shapefile instead of the original polygon P itself. This process is repeated for all extracted polygons, resulting in a final shapefile with the simplified output. This approach can be beneficial for reducing the complexity of the shapefile while preserving important geometric characteristics of the original polygons.

Fig. 3 illustrates the impact of the three geometric post-processing steps on the raw shapefile. These steps include deleting small polygons, deleting polygons inside another one, and simplifying over-segmentation.

In Fig. 4, a comparison is made among the raw output, the output after applying the Douglas-Peucker algorithm, and the output after applying the Pocket-based simplification algorithm. The differences among these outputs showcase how each algorithm affects the shapefile in terms of simplification and preservation of geometric details. This comparison can provide insights

into the effectiveness of each algorithm in achieving the desired level of simplification while maintaining the essential characteristics of the original data.

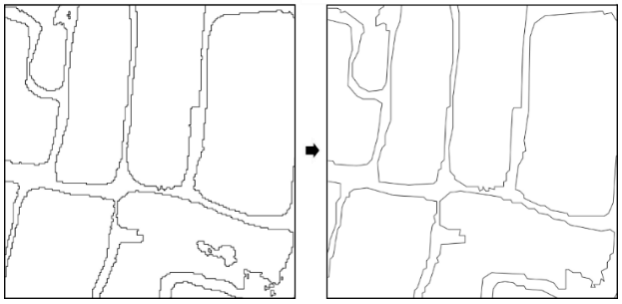

Fig. 3: Raw output (right) and the output after applying geometric post process (left), Simplified by Douglas-Peucker algorithm.

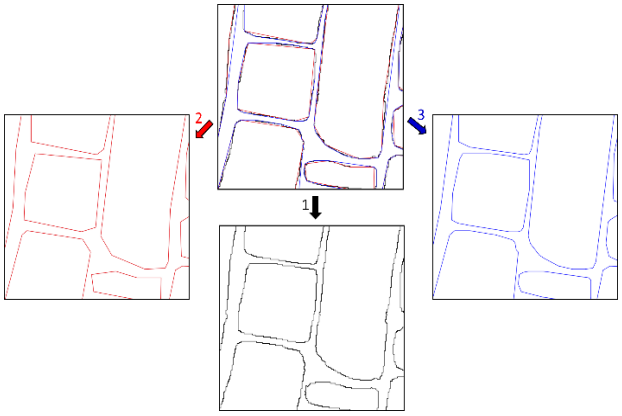

Fig. 4: Picture of boundary without simplification (1-black), simplified by Douglas-Peucker algorithm (2-red) and simplified by Pocket-based algorithm (3-blue).

Experimental Study

In this section, we present the data for training and how we prepare them to be ready for input to Mask R-CNN and the system process executed on.

A. Training Data

We use three images for training the network: an aerial image, an ortho image, and a Famenin Irrigated image. These images have dimensions of 4963 × 2819, 3999 × 3999, and 5520 × 3776 pixels respectively. The Famenin Irrigated image is a satellite image obtained from Google Earth, while the other two are UAV images taken in farmlands in Iran and Ethiopia. Since we are solving an instance segmentation problem, we need to create masks for each image to achieve our goal. We created the masks ourselves, as no online dataset satisfied our needs. To create the masks for the training data, we used LabelMe, a free graphical image annotation tool written in Python with a graphical interface built using Qt. In the final steps, each image was divided into tiles of 400 × 400 pixels, resulting in almost 300 tiles for the training data. Fig. 5 shows the result of a field together with its labeled mask. The only important requirement for the masks is that fields sharing the same boundary should not have the same color. Therefore, non-neighbor fields are free to have the same colors.

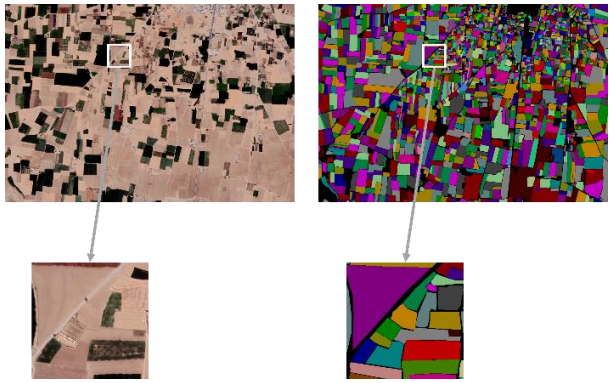

Fig. 5: Result of a training image with it's labeled mask.

B. System Config

The entire process of this study was implemented in Python. The training of the instance segmentation model was conducted using four K80 GPUs to optimize the balance between computational power and resource accessibility. This configuration facilitated a distributed training approach, enhancing efficiency and reducing training duration without compromising accuracy. The training time in this system takes approximately 2 hours.

Evaluation

In this section, we present the performance of our detected boundary.

A comparison of the accuracy assessment results obtained in this study with those from other studies cannot be done due to several reasons. Various UAVs and satellites may yield different quality images, and the study objects exhibit significant variations in terms of nature, size, location, and characteristics. To enable a reliable comparison of accuracy assessments across different feature extraction methods, each method needs to be individually studied and subsequently tested within the same study area(s) [17].

We calculate all accuracy assessments on one of the pictures from Famenin, which is 3432 × 3621 pixels. This picture isn't used for training and is only used for testing. For validating the predicted results with ground truth, a certain buffer is often considered for the reference boundary in cadastral mapping [19]. We consider both 5- and 6-pixel buffers for the reference boundary, which means the original reference boundary has a width of 1 pixel while in evaluation it is 5 or 6 pixels.

The measures we use to evaluate our work are common ones in this field: precision, recall, and F-score.

Precision represents whether the assigned boundary is valid. Recall shows the ability of the network to find all the boundaries, and F-score is the harmonic mean between precision and recall, which is a good overall measurement for final evaluation.

After overlapping our output boundary with the buffered reference, we calculate the mentioned measures using these formulas:

$$Precision = \frac{TP}{TP+FP} \quad (1)$$

$$Recall = \frac{TP}{TP+FN} \times BF \quad (2)$$

$$F-score = \frac{2 \times Precision \times Recall}{Precision+Recall} \quad (3)$$

If we consider binary classification for the confusion matrix, class positive stands for pixels labeled as "boundary," and class negative stands for pixels labeled as "non-boundary." Thus, we can create a confusion matrix and calculate the measures. BF in the recall formula stands for the buffer that we considered for the boundary reference in the Famenin test picture, and its quantification is different in each case. The rationale for the recall formula is that the sum of True Positives (TP) and False Positives (FP) represents the total number of detected boundary pixels, while TP + False Negatives (FN) indicates the total boundaries in the buffered reference, not the original reference. Therefore, to obtain the total boundaries in the original reference, which has a width of one pixel, we divide the sum of TP and FN by BF [6].

Results and Discussion

In this section, we present the performance of our detected boundary.

As we can see in Table 1, it is obvious that when we make the buffer thicker by 1 pixel, both precision and recall rise as more detected boundaries fall into the reference boundary buffer. A recall of 95% in a 6-pixel buffer shows that R-CNN finds almost all boundaries of the image. However, because the precision is less than 80%, we should note that some boundaries are detected that are not in the reference, making them extra.

If we fix the buffer, then the recall in both methods of simplifying polygons is almost equal, but precision is always better in the Pocket-based algorithm, indicating that more valid boundaries fall into the reference buffer in our method than in the Douglas-Peucker algorithm.

The important point is that precision achieved by a 5-pixel buffer and the paper's method is 67%, while this precision is achieved by the Douglas-Peucker algorithm only when the buffer is 6 pixels. This demonstrates how much stronger the Pocket-based algorithm works in finding valid boundaries compared to the Douglas-Peucker algorithm.

Based on better precision and almost equal recall in the Pocket-based simplification algorithm, our method consistently achieves a higher F-score. Therefore, based on these results, we can conclude that our simplification method, also known as the Pocket-based simplification algorithm, performs better in all tested situations.

Table 1: Accuracy comparison with respect to buffer and simplification method

Buffer	Method	Precision	Recall	F-Score
5 pixels	Douglas-Peucker	60	85	70
5 pixels	Pocket-based	67	87	76
6 pixels	Douglas-Peucker	66	95	78
6 pixels	Pocket-based	72	95	82

It is worth mentioning that our primary objective is to detect cadastral boundaries of farmlands. However, our presented method can also identify boundaries of buildings and other types of land in rural or urban areas. Additionally, there are many smallholders in our images, and detecting them is more challenging than identifying larger objects. One of the main issues with our detected boundaries is that the extracted polygons tend to be more rounded at the edges rather than straight. Therefore, geometric post-processing significantly impacts our network output and enhances its quality.

Conclusion

In this study, the Mask R-CNN model was used to solve instance segmentation for the automatic detection of cadastral boundaries in VHR images. The Mask R-CNN model is based on transfer learning, utilizing a pre-trained ResNet-50 backbone from the ImageNet dataset. The network can accept input at any desired resolution, and the output is a large mask of the input image, where values close to one represent the field. To extract the boundaries of each field from the masks, we apply Otsu's binarizing method to adaptively binarize the probability masks. Subsequently, the Canny edge detection algorithm is applied to the output, resulting in an image containing only cadastral boundaries. Since the instance segmentation method is used, the network's output contains individual boundaries for each field. Following the creation of a shapefile from the binary PNG output of the network, three geometric post-processing procedures are applied to enhance the raw output of the Mask R-CNN.

In the first step, polygons with an area smaller than a specific threshold based on the minimum area of farms in that region and the image's GSD are removed. In the second step, all polygons that are contained within another polygon are eliminated, as farms are non-overlapping. The final step involves using two methods to simplify the detected boundaries. While the Douglas-

Peucker algorithm, a well-known approach for line simplification, is utilized, a new method called the Pocket-based simplification algorithm is introduced and shown to outperform the Douglas-Peucker algorithm. This method is named after the pockets produced when creating the convex hull of each polygon. Based on a specific threshold, decisions are made regarding whether to add a pocket or connect edges between two endpoints of a polygon, resulting in a new simplified polygon. The Pocket-based simplification algorithm demonstrates higher accuracy in precision compared to the Douglas-Peucker algorithm, with nearly identical recall rates. This indicates that both methods can effectively identify cadastral boundaries, but the Pocket-based simplification algorithm excels at detecting valid boundaries. The final evaluation is based on the F-score, which is the harmonic mean between precision and recall and consistently favors the Pocket-based simplification algorithm over the Douglas-Peucker algorithm.

Instance segmentation for detecting automatic cadastral boundaries has shown promising results, suggesting that employing other networks optimized for instance segmentation may further improve outcomes.

There are some open problems:

Our algorithm does not detect and delete urban/rural areas, resulting in segmentation of those areas, which reduce the precision and recall of our algorithm. One solution is to use semantic segmentation to detect and delete urban/rural areas. Then, we use our algorithm to detect boundary of farmlands.

The result heavily depends on the texture and shape of farmland that differs in different geographical regions and type of farmlands.

In this research, we utilize only the three visible channels of images (RGB). However, incorporating invisible bands alongside RGB bands in satellite images can enhance the performance of this algorithm.

Abbreviations

VHR	Very Hight Resolution
UAV	Unmanned Aerial Vehicle
EO	Earth Observation
MRS	Multi Resolution Segmentation
CNN	Convolutional Neural Network
MCG	Multiresolution Combinatorial Grouping
FCN	Fully Convolutional Network

References

[1] X. Luo, R. M. Bennett, M. Koeva, C. Lemmen, "Investigating semi-automated cadastral boundaries extraction from airborne laser scanned data," Land, 6(3): 60, 2017.

[2] I. Williamson, "The justification of cadastral systems in developing countries," Geomatica, 51: 21-36, 1997.

[3] S. Enemark, K. C. Bell, C. Lemmen, R. McLaren, Fit-for-purpose land administration. International Federation of Surveyors (FIG), Copenhagen: International Federation of Surveyors (FIG), 2014.

[4] X. Luo, R. Bennett, M. Koeva, C. Lemmen, N. Quadros, "Quantifying the overlap between cadastral and visual boundaries: A case study from Vanuatu," Urban Sci., 1: 32, 2017.

[5] I. Williamson, S. Enemark, J. Wallace, A. Rajabifard, Land administration for sustainable development, Citeseer, 2010.

[6] X. Xia, C. Persello, M. Koeva, "Deep fully convolutional networks for cadastral boundary detection from UAV images," Remote Sens., 11(14): 1725, 2019.

[7] A. L. Dr Samantha Lavender, Trusted Earth Observation Experts, 2012.

[8] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, F. Fraundorfer, "Deep learning in remote sensing: A comprehensive review and list of resources," IEEE Geosci. Remote Sens. Mag., 5: 8-36, 2017.

[9] J. R. Bergado, C. Persello, C. Gevaert, "A deep learning approach to the classification of sub-decimetre resolution aerial images," in Proc. 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016.

[10] D. Garcia-Gasulla, F. Parés, A. Vilalta, J. Moreno, E. Ayguadé, J. Labarta, U. Cortés, T. Suzumura, "On the behavior of convolutional nets for feature extraction," J. Artif. Intell. Res., 61: 563-592, 2018.

[11] L. Meyer, F. Lemarchand, P. Sidiropoulos, "A deep learning architecture for batch-mode fully automated field boundary detection," Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., 43: 1009-1016, 2020.

[12] S. Crommelinck, R. Bennett, M. Gerke, F. Nex, M. Y. Yang, G. Vosselman, "Review of automatic feature extraction from high-resolution optical sensor data for UAV-based cadastral mapping," Remote Sens., 8(8): 689, 2016.

[13] L. Drăguţ, O. Csillik, C. Eisank, D. Tiede, "Automated parameterisation for multi-scale image segmentation on multiple layers," ISPRS J. Photogramm. Remote Sens., 88: 119-127, 2014.

[14] Y. Li, S. Wang, Q. Tian, X. Ding, "A survey of recent advances in visual feature detection," Neurocomputing, 149: 736-751, 2015.

[15] S. Crommelinck, R. Bennett, M. Gerke, M. Y. Yang, G. Vosselman, "Contour detection for UAV-based cadastral mapping," Remote Sens., 9: 171, 2017.

[16] S. Crommelinck, M. Koeva, M. Y. Yang, G. Vosselman, "Application of deep learning for delineation of visible cadastral boundaries from remote sensing imagery," Remote Sens., 11: 2505, 2019.

[17] B. Fetai, K. Oštir, M. Kosmatin Fras, A. Lisec, "Extraction of visible boundaries for cadastral mapping based on UAV imagery," Remote Sensing, 11: 1510, 2019.

[18] J. Wang, J. Song, M. Chen, Z. Yang, "Road network extraction: A neural-dynamic framework based on deep learning and a finite state machine," Int. Remote Sens., 36: 3144-3169, 2015.

[19] D. Poursanidis, N. Chrysoulakis. Z. Mitraka, "Landsat 8 vs. Landsat 5: A comparison based on urban and peri-urban land cover mapping," Int. J. Appl. Earth Obs. Geoinf., 35: 259-269, 2015.

[20] C. Persello, A. Stein, "Deep fully convolutional networks for the detection of informal settlements in VHR images," IEEE geoscience and remote sensing letters, 14: 2325-2329, 2017.

[21] Y. Xu, Z. Zhu, M. Guo, Y. Huang, "Multiscale edge-guided network for accurate cultivated land parcel boundary extraction from remote sensing images," IEEE Trans. Geosci. Remote Sens., 62, 2023.

[22] Z. Cai, Q. Hu, X. Zhang, J. Yang, H. Wei, J. Wang, Y. Zeng, G. Yin, W. Li, L. You, B. Xu, Z. Shi, "Improving agricultural field parcel delineation with a dual branch spatiotemporal fusion network by

integrating multimodal satellite data," ISPRS J. Photogramm. Remote Sens., 205: 34-49, 2023.

[23] M. T. Metaferia, R. M. Bennett, B. K. Alemie, M. Koeva, "Furthering automatic feature extraction for fit-for-purpose cadastral updating: Cases from Peri-Urban Addis Ababa, Ethiopia," Remote Sensing, 15: 4155, 2023.

[24] A. M. Hafiz, G. M. Bhat, "A survey on instance segmentation: state of the art," Int. J. Multimedia Inf. Retr., 9: 171-189, 2020.

[25] R. Girshick, J. Donahue, T. Darrell, J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2014.

[26] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, M. Pietikäinen, "Deep learning for generic object detection: A survey," Int. J. Comput. Vision, 128: 261-318, 2020.

[27] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012.

[28] K. E. A. Van de Sande, J. R. R. Uijlings, T. Gevers, A. W. M. Smeulders, "Segmentation as selective search for object recognition," in Proc. 2011 International Conference on Computer Vision, 2011.

[29] R. Girshick, "Fast R-CNN," in Proc. the IEEE International Conference On Computer Vision: 1440-1448, 2015.

[30] S. Zagoruyko, A. Lerer, T. Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, P. Dollár, "A multipath network for object detection," arXiv preprint arXiv: 1604.02135, 2016.

[31] S. Ren, K. He, R. Girshick, J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," Advances in neural information processing systems, 28, 2015.

[32] K. He, G. Gkioxari, P. Doll'ar, R. Girshick, "Mask r-cnn," in Proc. the IEEE International Conference on Computer Vision, 2017.

[33] K. Wada, labelme: Image Polygonal Annotation with Python, GitHub, 2018.

[34] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Proc. 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[36] J. D. Yang, Y. S. Chen, W. H. Hsu, "Adaptive thresholding algorithm and its hardware implementation," Pattern Recognit. Lett., 15: 141-150, 1994.

[37] J. Canny, "A computational approach to edge detection," IEEE Transactions on pattern analysis and machine intelligence, PAMI-8(6): 679-698, 1986.

[38] M. De Berg, Computational geometry: algorithms and applications, Springer Science & Business Media, 2000.

Biographies

Neda Rahimpour Anaraki received her B.Sc. degree from Alzahra University, and her M.S. degree from Amirkabir University of Technology in 2017 and 2019, respectively, both in Computer Science. She is currently Ph.D. candidate at Shahid Beheshti University. Her research interests are computational geometry, remote sensing, computer vision, machine learning and deep learning.

- Email: neda.rpa@gmail.com
- ORCID: 0009-0000-4382-7145
- Web of Science Researcher ID: NA
- Scopus Author ID : NA
- Hompage: NA

Alireza Azadbakht earned his B.Sc. in Computer Science from Shahid Beheshti University between 2016 and 2020, followed by an MSc, specializing in data mining, from the same institution between 2020 and 2022. His core research interests lie in computer vision, deep learning, remote sensing, data science, and machine learning.

- Email: ali.r.azadbakht@gmail.com
- ORCID: 0000-0002-9663-8388
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Hompage: NA

Maryam Tahmasbi received her Ph.D. in computer science from Amirkabir university of technology 2009. She is currently assistant professor in department of computer and data sciences in Shahid Beheshti university. Her research interest is geometric data analysis concentrating on image processing.

- Email: m_tahmasebi@sbu.ac.ir
- ORCID: 0000-0001-9513-0251
- Web of Science Researcher ID: NA
- Scopus Author ID: Maryam Tahmasbi - 56120848900
- Hompage: https://mathsci.sbu.ac.ir/~m_tahmasbi

Hadi Farahani is an assistant professor in computer science at Shahid Beheshti University. He received his Ph.D. in mathematics from Shahid Beheshti University in 2011. His research interests includes machine learning, deep learning, blockchain technology and applications of logic in computer science.

- Email: h_farahani@sbu.ac.ir
- ORCID: 0000-0002-8067-0383
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Hompage: https://mathsci.sbu.ac.ir/~h_farahani

Saeed Reza Kheradpishe is an assistant professor in computer science at Shahid Beheshti University. He received his Ph.D. in computer science from the University of Tehran in 2017. His research interests include machine learning, deep learning, spiking neural networks and computational neuroscience.

- Email: s_kheradpishe@sbu.ac.ir
- ORCID: 0000-0001-6168-4379
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Hompage: https://mathsci.sbu.ac.ir/~s_kheradpisheh

Alireza Javaheri received his B.Sc. degree and M.S. degree from Shahid Beheshti University, Tehran, Iran, respectively, both in Computer Science. His research interests are in the areas of Machine Learning, Deep Learning and Meta-Learning.

- Email: alireza.jvh98@gmail.com
- ORCID: 0000-0002-9663-8388
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Hompage: NA

How to cite this paper:
N. Rahimpour Anaraki, A. Azadbakht, M. Tahmasbi, H. Farahani, S. R. Kheradpishe, A. Javaheri, "Automatic cadastral boundary detection of very high resolution images using mask R-CNN," J. Electr. Comput. Eng. Innovations, 12(2): 525-534, 2024.

DOI: 10.22061/jecei.2024.10650.727

URL: https://jecei.sru.ac.ir/article_2151.html