Detection of Fairy Circles in UAV Images Using Deep Learning

Yuhong Zhu[1], Zahra Moayed[1], Barbara Bollard-Breen[2], Ashray Doshi[2],
Jean Baptiste Ramond[3], and Reinhard Klette[1]
[1] Department of Electrical and Electronic Engineering
School of Engineering, Computer, and Mathematical Sciences
Auckland University of Technology, New Zealand
{ yrk8742,zmoayed,rklette}@aut.ac.nz
[2] Institute for Applied Ecology New Zealand, School of Science
Auckland University of Technology
bbreen@aut.ac.nz
[3] The Centre for Microbial Ecology and Genomics
University of Pretoria, South Africa
jbaptiste.ramond@gmail.com

Abstract

Fairy circles are circular patches of barren soil forming large clusters in the arid grasslands of Southern Africa (especially in Namibia) and Western Australia. Fairy circles are clearly visible in aerial images shown in applications such as Google Maps, and they can be recorded using sensors mounted on drones in very high image and video resolution for ecological studies aiming at understanding the origin of these patterns. Traditional analysis of fairy circles is done by manual digitising and counting. We also showed recently that, despite being challenging, traditional computer vision methods enabled the detection of fairy circles. To improve fairy circle detection and localization automatically in aerial images, we here present the use of a convolutional neural network (CNN). The results suggest that new methods using CNNs outperform other methods in terms of accuracy.

I. Introduction

Fairy circles are still one of the unsolved natural mysteries of our planet. Until their recent description in the Australian outback [5], fairy circles were thought to be endemic to the Namib desert dune and gravel plain. Image data has been used to analyse their spatial distribution shape, area, distribution and to estimate their lifespan [8][1][4]. Satellite imagery, obtained with Google map and/or Apple map, clearly show fairy circles at low resolution. In contrast, *Unmanned aerial vehicles* (UAVs or drones) can capture high spatial and spectral resolution imagery depending on the sensor used. And, once the imagery has been processed into orthomosaics, fairy circles are generally manually digitised as their automatic detection has mainly been unsuccessful [1].

Previous computer vision publication aiming at detecting FCs, have followed the traditional approach of image processing, segmentation, and subsequent classification of the obtained segments [1]. These methods not only detect FCs but also can draw their contours. The results obtained were evaluated based in the comparisons of *miss rate* and *false-positives per image*. They showed that the performance these detection methods were site-dependent and were also influenced by lighting conditions and image resolution. In short, automatic fairy circle detection is not yet accurate, scale-able or repeatable.

Due to the complex environmental and ecological conditions that give rise to FCs, as well as the fact that they occur on different soil substratum (e.g., dune and gravel plain in the Namib Desert [10]), their appearance on digital images can vary in colour, shape, size or distribution.

Furthermore, in Namibia, the contrast between soil and the plants is low as they typically present similar color, brightness and texture. The scattered plants spread in the *background* (i.e. the area around the fairy circles) also increases the noise for detection. These are primary obstacles especially when using distance transform, aerial segmentation techniques, and other traditional detection methods. Furthermore, the existing research used a filter

TABLE I. *Top*. Structure of ConvNet, where k denotes kernel size ($width \times height \times \#channels$). *Bottom*. Structure of fully-connected layer.

	Layer	k	Output size	Receptive field size
1	conv1	$3 \times 3 \times 64$	$40 \times 40 \times 64$	3×3
2	pooling1	2×2	$20 \times 20 \times 64$	4×4
3	conv2	$1 \times 1 \times 32$	$20 \times 20 \times 32$	4×4
4	conv3	$3 \times 3 \times 32$	$20 \times 20 \times 32$	8×4
5	pooling2	2×2	$10 \times 10 \times 32$	10×10
6	conv4	$3 \times 3 \times 64$	$10 \times 10 \times 64$	18×18
7	pooling3	2×2	$5 \times 5 \times 64$	22×22

	Layer	Input	Output
1	FC1(input)	1600	256
2	FC2(hidden)	256	128
3	FC3(output)	128	3

of fixed size for FC detection in the last step, and this can be improved by adding a stronger "reasoning procedure" like a logic layer for further judgment.

II. Proposed Method

This section describes the steps that were used for detecting FCs in this study.

Data Preparation. For a deep learning approach the initial amount of data must be significant. The samples used in this study are high resolution RGB images taken by UAVs at 2 sites in Namibia. Firstly, FCs were manually labeled and a new contour binary image was generated in the orthomosaic. For further processing, the orthomosaic was then resized into 707×643 and the original RGB data was extracted. Since every orthomosaic was generated from geotagged images, the exact location of each FC was known.

Figure 1 shows the general steps performed to extract and prepare data. In total, 445 samples from the dune site and 284 from the gravel plain one have been collected. To increase the accuracy, data augmentation was implemented by doing horizontal reflections and rotation and 45 degree rotation of each image 8 times. As a result, at the end of this process, our dataset comprised nearly 9000 images, i.e., was enhanced 16 times.

Recognition of Fairy Circles. We used a deep learning approach to detect FCs. The proposed network in this paper used 40×40 gray-scale images as input and had four convolutional layers followed by three fully connected layers. All convolutional layers used a 3×3 kernel with 'SAME' padding. Max pooling layers used a kernel size of 2×2, and stride 2. Table I shows the network structure.

To avoid overfitting, L_2 regularisation was applied for the first two fully connected layers. The network simply returned the likelihood of the presence of an object (i.e., FC, ground, track) in a given image. This process is illustrated in Fig. 2. When using a 500 images dataset, the average accurate prediction of the presence of an FC following this approach was 81.9%. When ignoring likelihoods below 50%, the average likelihood increased to 92.9% with an accuracy of 85%. An likelihood threshold of 85% was chosen based on these reported values to validate FC detection.

Detection of Fairy Circles. A *convolutional neural network* (CNN) is mainly designed to classify objects. Further processing is needed to detect and localise them. For this purpose, we applied two general methods; sliding windows and selective search.

Sliding Window. We used a notation based on two points to describe the sliding window: (x_1, y_1) and (x_2, y_2) denote the top-left and bottom-right corners of the window, respectively. While the window slides in the image of size $w \times h$.

In the 'normal' sliding window method, the window starts sliding from a location such that the top-left corner of the window (x_1, y_1). The total number of window positions is calculated as below

$$\lceil \frac{w \cdot l}{s} \rceil \times \lceil \frac{h \cdot l}{s} \rceil \quad (1)$$

where s is the step size and $l = x_2 - x_1$. Ultimately, the remaining areas which were located around the image's border and that do not fill completely the sliding window were ignored.

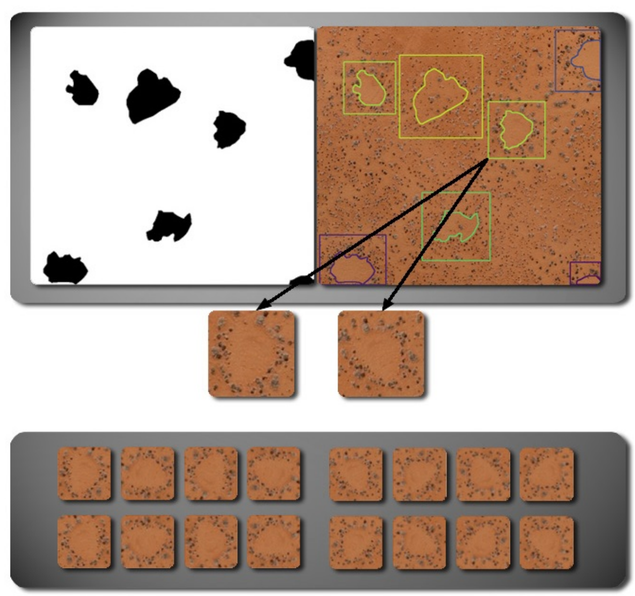

Fig. 1. Procedure of sample extraction and augmentation. *Upper Left*. Ground truths of FCs, converted to binary images. *Upper Right*. RGB image of corresponding FCs. Sub-images in purple bounding boxes were not used as they were not complete or centered. *Middle Left*. Sub-image extracted from RGB image. *Middle Right*. The same image as on left side but rotated 45° for data augmentation. *Lower*. Each image was flipped once and then rotated by 90° for three times.

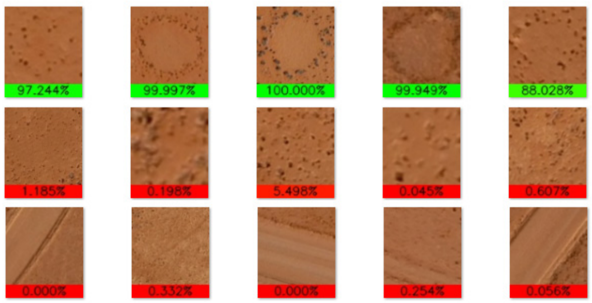

Fig. 2. Likelihood that an image shows an FC as calculated by the proposed CNNs.

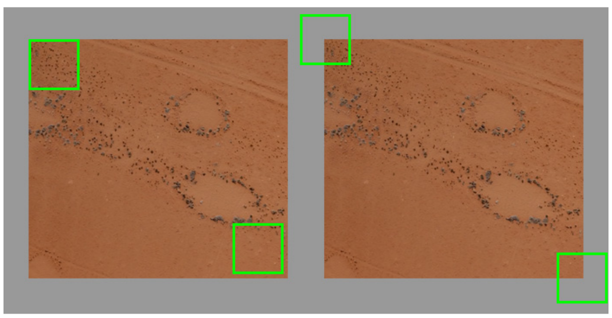

Fig. 3. Start (top-left) and end (bottom-right) locations of a fixed-size sliding window. *Left*. Normal sliding window. *Right*. Full sliding window.

The 'full' sliding window method solved the aforementioned edge problem, as well as the detection problems encountered when increasing the numbers of windows. The differences between the 'full' and the 'normal' sliding window methods are their initial and end positions, as shown in Figure 3. In the 'full' method, we only considered the image area completely inside the sliding window. A non-square image at an edge and a smaller square at a corner were uniformly mapped into the same input size of 40×40. In this way, $\lceil w/s \rceil \times \lceil h/s \rceil$ sub-images could be collected in a $w \times h$ image.

Selective Search. A sliding window approach may scan the entire area of an image, however, it is time consuming. Inspired by Region-Based CNN (R-CNN) [6] which uses selective search [9] to generate region proposals, we designed a similar method to produce potential objects. We then used a region-proposal method, derived from [1], to replace the sliding windows and to reduce the computation time.

The selective search procedure to detect potential FCs thus started with a non-local means denoising algorithm [2] aiming at smoothing areas inside the FCs (see Fig. 4), followed by a series of Gaussian and Laplacian operations, distance transforms and thresholding. We used three Gaussian kernels of sizes 3×3, 7×7, and 11×11 to extract different sizes. The bounding boxes around FCs

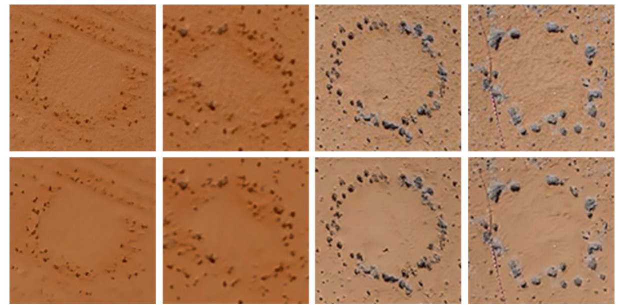

Fig. 4. *Top*. FCs before denoising. *Bottom*. FCs after denoising.

were then filtered by size. For eligible boxes, centre points were labelled to create a centre map. The purpose of this stage was to increase true positives while false positives were further reduced using a CNN.

Refinement of Bounding Boxes. The detection of potential FCs with the approaches described above may lead to the observation of duplicated bounding boxes for one object. Also, due to FCs' appearance, multiple objects might be merged in one bounding box. In many algorithms, bounding boxes are confirmed by adding anchor box and offset as outputs in CNNs and by then calculating Intersection over Union (IOU). In this paper, we used a simple method which works on the centre point generated in the previous detection step.

As the size of an input image is much larger than the CNNs input (i.e. it may contain many FCs), it is first demagnified from 707×643 to 303×321. We then applied a series of sliding windows with side lengths of 100, 80, and 60, and a step size of 5, to detect potential FCs. A threshold 85% was used to filter the results and confirm their corresponding location. Possible bounding boxes were defined by using these locations.

The sliding window step provided with the centres of all the bounding boxes. To allocate those centres to their corresponding FC, grouped them based on their distance to one another. A centre is grouped by the eight centre points surrounding it (four corners and four edges), and it is separated from other centres. So, the expected distance between two centres is larger than the distance from a centre to its corner points but smaller than twice the distance to edge centres (between $\sqrt{2}s$ and $2s$ where s is the step size of the sliding window). We then use

$$r = \lceil \frac{\sqrt{2}}{2} \cdot s \rceil \tag{2}$$

as the radius to draw a solid circle on top of all centres on a new image, called the *centre map*. The radius was 4 when we set a step size of 5. Subsequently, centres with a distance smaller than 8 will form a larger region on the centre map; while isolated centres will be just small regions. We assumed that the larger regions were potential

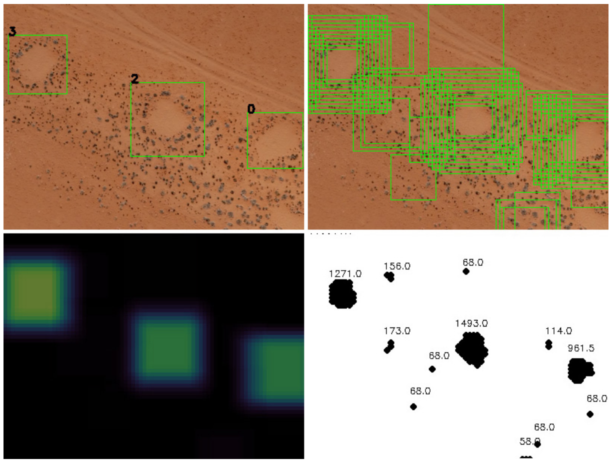

Fig. 5. *Upper left*. Final bounding boxes. *Upper right*. All the potential areas defined by multiple sliding windows. *Lower left*. A heat map based on the candidate areas; a color map is applied to the grayscale image. *Lower right*. A centre map created by the sliding windows; it shows all the centres of bounding boxes from the sub-image above

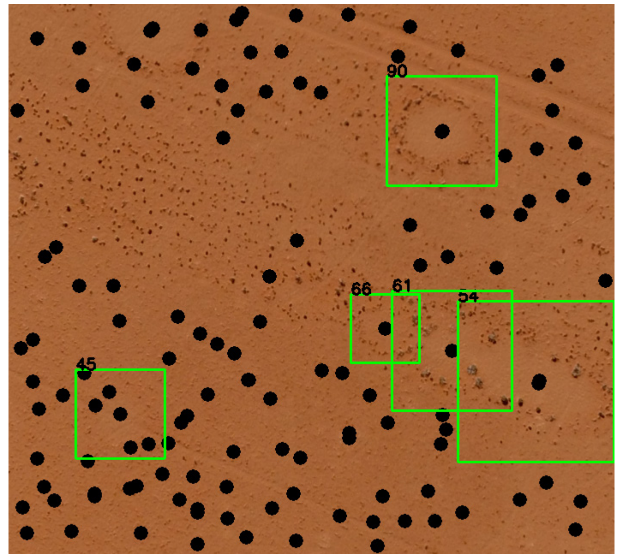

Fig. 6. Result of the selective search method. Black areas are centres of candidate bounding boxes, final bounding boxes are shown in green.

FCs, and used their centroids as the real FCs' centres. Small regions were further ignored.

Once a centroid was confirmed, a set of square bounding boxes (which use the centroid as their centre) were placed on the image; those boxes vary in side lengths for matching the given size of an extracted sample. These samples were fed to the network again to evaluate the probability of presence of the FC. Non maximum suppression was used to choose a unique bounding box.

The localisation algorithm, using selective search, was similar to the one used for the sliding window, but without filtering out smaller regions in the centre map.

III. Experimental Results

To compare different approaches, we used online images [3], [11] of FC fields that we never used when training the model. They were cut into 800×600 pixels and the ratio of the size of FCs to the image size remained in the range of $(2\%, 10\%)$ so that an FC would fit into a sliding window.

Figures 5 and 6 illustrate the output results of the sliding window and the selective search methods, respectively.

For a quantitative comparison, two measures, *precision* (PR) and *recall* (RC), were used to assess the performance. PR denotes the ratio of numbers of true-positive to all detections, and RC is the ratio of numbers of true-positives and of all ground-truth FCs:

$$PR = \frac{tp}{tp + fp} \tag{3}$$

and

$$RC = \frac{tp}{tp + fn} \tag{4}$$

where tp, fp and fn are the numbers of true-positives, false-positives, and false-negatives, respectively.

Sliding Window. Table II shows a comparison between two sliding window methods. We selected ten images which all had truncated FCs on the edge. The results suggested that by extending the sliding window method from normal to full coverage, the average PR and RC values increased by 0.09 and 0.16, respectively. Figure 7 shows that the full sliding window method leads to the detection of more centroids at the border of the centre map. Although the normal sliding window can cover almost entirely of an image, the centres of sliding windows in this method do not cover the image-border area. The improved full sliding window algorithm significantly increased the detection accuracy near the image border. For higher accuracy, we only considered this improved methods afterwards.

Selective Search. Figure 8 illustrates the difference between traditional selective search and the discussed modified version for detecting FCs. It clearly showed that the method we propose is more accurate.

To compare with existing methods, we used the same dataset as in [1]. Table III compares the detection accuracy of five different methods.

A drawback of the three methods used in [1] is the manual setting of parameters according to the different environments studied. Consequently, these methods fail when operating on new images. Furthermore, small FCs may be ignored by these approaches as they implement a size-based filtering step.

The method using new version of selective search as region proposal can be seen as a combination of methods proposed in [1] and CNNs where CNN reduces false

TABLE II. Comparison between full and normal sliding window methods

Image	Ground truth	Full				Normal			
		TP	FP	PR	RC	TP	FP	PR	RC
I_1	32	19	0	1	0.59	17	0	1	0.53
I_2	25	19	0	1	0.76	14	0	1	0.56
I_3	21	13	0	1	0.62	15	0	1	0.71
I_4	21	16	0	1	0.76	16	0	1	0.76
I_5	17	14	2	0.88	0.82	13	1	0.93	0.76
I_6	5	3	0	1	0.60	3	0	1	0.6
I_7	7	5	2	0.71	0.71	4	1	0.8	0.57
I_8	5	5	1	0.83	1	4	1	0.8	0.8
I_9	4	2	0	1	0.50	2	0	1	0.5
I_{10}	1	1	0	1	1	0	0	0	0
Average				**0.94**	**0.74**			**0.85**	**0.58**

TABLE III. Comparison between three methods used in [1] with the full sliding window and selective search, as proposed in this paper

Image	Method A		Method B		Method C		Full Sliding Window		Selective Search	
	PR	RC	PR	RC	PR	RC	PR	RC	PR	RC
I_1	0.67	0.8	1	0.8	1	0.8	1	0.6	1	0.8
I_2	0.5	1	0.67	0.8	0.6	0.6	1	0.8	0.83	1
I_3	0.67	1	0.8	1	0.8	1	1	1	0.8	1
I_4	0.5	0.5	0.6	0.75	0.6	0.75	1	0.5	1	0.5
I_5	0.8	1	.43	0.75	0.4	0.5	1	0.75	0.75	0.75
Average	**0.63**	**0.86**	**0.7**	**0.82**	**0.68**	**0.73**	**1**	**0.73**	**0.88**	**0.81**

positives. Selective search still lead to the observation of false positives; while after CNN's filtering, PR value increases by almost 0.2 apposed to methods proposed in [1] but keeps a similar RC.

There are two key aspects that influence the performance of the selective search method. First, similar to previous methods in [1], the selective search method still cannot overcome all the variations in environments, although it can cope with more diversity. This can be clearly seen when applying the method on some gravel plain sets (see Fig. 9). Second, CNNs also need accurate input data with deep structure and numerous weights. Otherwise, CNNs cannot filter the candidate areas properly. Without an area limit, as for the sliding window method, our CNNs have a higher chance to return false positives, which decreases the PR when compared to the sliding window method.

Table IV shows comparative results for sliding window methods using step sizes of 5 and 10, which have a higher accuracy than the selective search method. The comparison uses multiple test sets. The NAMIBIA dataset is the same one used in Table III. The ONLINE dataset

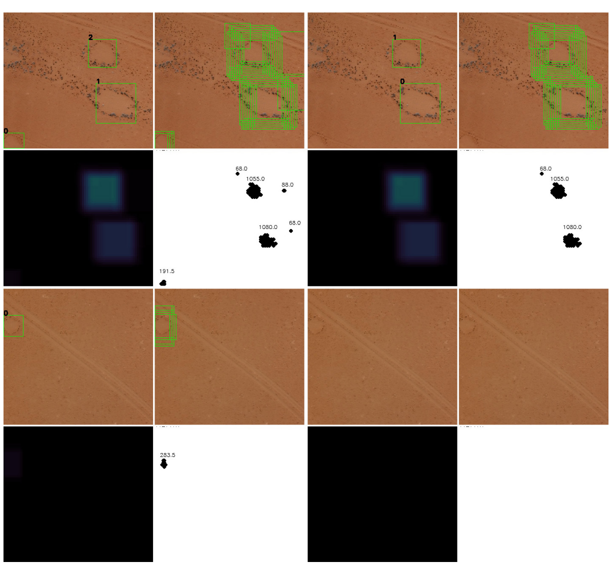

Fig. 7. Full *Left* and normal *Right* sliding window

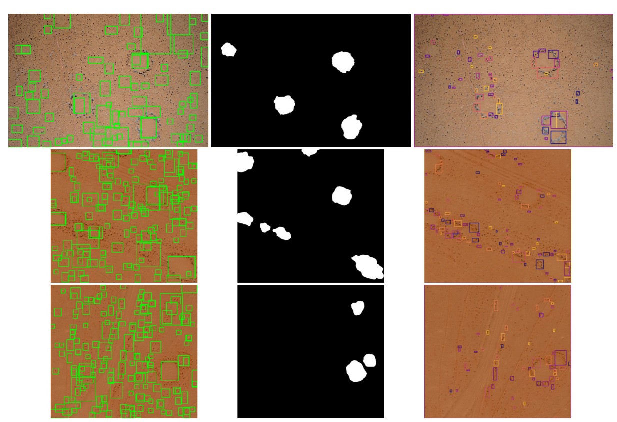

Fig. 8. *Left*. Modified selective search for FCs. *Middle*. Ground truth. *Right*. Traditional selective search.

TABLE IV. Comparison between sliding window method with different step sizes. Average is a weighted arithmetic mean of PR and RC

Test Set	Sliding Window s5		Sliding Window s10	
	PR	RC	PR	RC
NAMIBIA×5	1	0.73	0.93	0.69
INTERNET×15	0.97	0.74	0.97	0.69
DUNE×20	0.86	0.83	0.89	0.77
GRAVEL×15	0.62	0.67	0.71	0.67
Average	**0.84**	**0.75**	**0.87**	**0.71**

has 15 processed images from online sources [3], [11]. We cut the original images into pieces, or rotated them to obtain different scales and numbers of FCs. We also left some truncated samples near the image border. The DUNE and GRAVEL datasets originate from our training sets of images from Namibia, and are characterised by distinct FC surroundings.

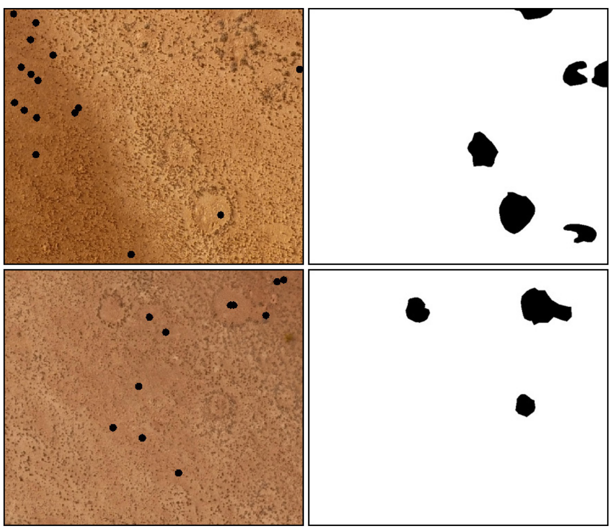

Fig. 9. Two images from a gravel site show that our selective search method does not cover FCs very well in these situations

When using the sliding window methods, a larger step size extracts fewer samples. It therefore reduced the computing time significantly from 35 to 15 seconds per image on an average laptop. Furthermore, our localisation method did not trust small areas on the centre map. So, it rejected more false-positives than true-positives and increased PR slightly but decreased RC at the same time.

IV. Conclusions

This paper presents two detection methods and one localisation method with a CNN as pattern recognition algorithm in order to detect fairy circles. A modified version of a selective search method was also used for presenting experimental results. Our results suggest that the sliding window method we propose outperforms the other methods tested while handling trade-offs between PR and RC. However, our selective search did not overcome all the challenges of environment variations.

The experiments clearly showed the possibility of obtaining higher accuracy by using CNNs. Regarding future work, further processes should be added to improve the drawing of FC contours and/or to estimate their surface area. Those processes would focus on the FCs' bounding boxes which are much smaller areas than the whole image. Furthermore, pixel-level CNN models [7] are considered to be another option for high detection and segmentation accuracy.

References

[1] Al-Sarayreh, M., Moayed, Z., Bollard-Breen, B., Ramond, J.-B., & Klette, R.: Detection and spatial analysis of fairy circles. In Proc. IEEE Image Vision Computing New Zealand, DOI: 10.1109/IVCNZ.2016.7804457, 2016.

[2] Antoni, B., Bartomeu, C., & Jean-Michel, M.: Non-local means denoising. Image Processing Online, 1:208–212, 2011.

[3] Fay, J. M.: The 'fairy circles' in Namibia [Photograph]. Retrieved from www.ecography.org/sites/ecography.org/files/styles/threshold-992/public/nationalgeographic1022083.jpg

[4] Getzin, S., Wiegand, K., Wiegand, T., Yizhaq, H., von Hardenberg, J., & Meron, E.: Adopting a spatially explicit perspective to study the mysterious fairy circles of Namibia. Ecography, 38(1), 1-11, 2015.

[5] Getzin, S., Yizhaq, H., Bell, B., Erickson, T. E., Postle, A. C., Katra, I., Tzuk, O., Zelnik, Y. R., Wiegand, K., Wiegand, T., Meron, E.: Discovery of fairy circles in Australia supports self-organization theory. PNAS, 113(13):3551–3556, 2016.

[6] Girshick, R., Donahue, J., Darrell, T., & Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. IEEE Trans. Pattern Analysis Machine Intelligence, 38(1):142–158, 2016.

[7] He, K., Gkioxari, G., Dollár, P., & Girshick, R.: *Mask R-CNN*. In Proc. ICCV, 2980–2988, 2017.

[8] Tschinkel, W. R.: The Life Cycle and Life Span of Namibian Fairy Circles. PloS one, 7(6), 2012.

[9] Uijlings, J. R. R., van de Sande, K. E. A., Gevers, T., & Smeulders, A. W. M.: Selective search for object recognition. Int. J. Computer Vision, 104(2):154–171, 2013.

[10] Van der Walt, A. J., Johnson, R. M., Cowan, D. A., Seely, M., & Ramond, J.-B.: Unique microbial phylotypes in Namib desert dune and gravel plain Fairy Circle soils. Applied Environmental Microbiology, doi:10.1128/AEM.00844-16, 2016.

[11] Vandyke, I.: Namibia's mysterious fairy circles dot the landscape at Sossusvlei [Photograph]. Retrieved from www.wildimages-phototours.com/wpcontent/uploads/2017/06/Sossusvlei-10-Namibia-Inger-Vandyke.jpg