Centre for Digital Built Britain

2018/19 ECR Project

Fly Safe: Aerial Swarm Robotics using Force Field Particle Swarm Optimisation

Final Reporting

10/07/2019

Name of ECR: Shan Luo	Affiliation: University of Liverpool
	Email address: shan.luo@liverpool.ac.uk
(if applicable)	Name of Research Assistant: Lauren Parker
PI name	

July 10, 2019

1 Abstract

This project proposes a solution to the problem of maintaining the infrastructure of bridges and other structures. New designs and technologies allow bridges to be built higher, reach longer, and span a river, valley or sea that were deemed impossible to cross before. This creates hazardous environments that are hard for humans to reach, making the labour intensive inspection and maintenance process ineffective. Such hazardous inspection and maintenance environments are not limited to bridges and exist across all forms of critical infrastructure. However, this task may have the potential to be done by autonomous systems. In this project we aim to create a coordinated aerial swarm system to inspect the cracks in the bridge structures, improving the monitoring coverage and efficiency. With strict constraints of the environments and mutual interference, the development of such multi-UAV system for bridge inspection is challenging. We have achieved the following objectives of the project: Agile coordination and collision avoidance for aerial swarms (WP1); Image based bridge defect detection and assessment (WP2); System integration and validation (WP3).

2 Main Text

2.1 Background

We reviewed related works describing the main ideas used in this project which covers Convolutional Neural Networks, Active Contours and image processing used in the field of crack detection. Crack detection is a widely researched process to identify cracks in different structures and surfaces using a variety of techniques. It is a vital check to ensure that a structure is maintained and continues to be safe; but it suffers from difficulties with the lighting or noise in an image. Some of the techniques used in crack detection that were the most relevant to this project are reviewed. We also review a few sources that examine aerial vehicle systems to detect cracks in infrastructure.

2.1.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a form of neural networks used in machine learning that can be used for image analysis and are known for semantic segmentation [1]. They are based on biological processes [2, 3] as the structure of neural networks resembles the arrangement of neurons in a brain.

CNNs have already been used in crack detection for their powerful ability to classify different images or parts of an image. CrackNet [4] uses a unique CNN where it has no pooling layers to retain the images size throughout training and has a high F-score value over 200 images. In [5] they use a CNN without any image processing techniques to classify regions of an image. This CNN configuration was used in this project and comes from a Github repository that implements it [6].

2.1.2 Active Contours

Active Contours (ACs) is a framework model in computer vision first introduced by Kass at. al [7]. They are sometimes referred to as snakes that deform according to an energy minimising function and can be used for edge detection, segmentation and shape recognition. Geodesic active contour (GAC) [8, 9] is an AC that utilises the idea of Euclidean curve shortening evolution. The snakes will evolve until they reach a threshold. A version of AC to emerge is Adaptive Active Contours, whereby the tuning of vital parameters are automatically adjusted in the algorithm [10, 11] or using an optimisation procedure [12] or using a weight matrix to force the curve evolution [13]. This was not obtainable in this project but is why several thresholding methods were looked at to compare values of the threshold. An implementation in [14] combines the AC model with support vector machines to eliminate noise. Morphological Snakes discussed in [15] are related to ACs but they use morphological operations such as dilation and erosion [16] instead of solving PDEs. Python has the image processing library scikit-image [17] that uses morphological snakes with GAC to identify contours in noisy images, providing they are pre-processed using for example an inverted gaussian filter.

We wanted to combine the power of CNNs with the edge detection abilities of the ACs which has been recognised as efficient and accurate [18–20]. They use the CNN and AC in parallel unlike our proposed solution which uses the CNN and AC seperately.

2.1.3 Image Processing

Image Processing techniques [21, 22] are used in a wide variety of fields of research but are very important in computer vision as real images are not perfect to identify shapes or objects, they contain noise or artefacts that we need to remove in order to work with the images. The technique of most interest is Morphological Operations [16], they are simple operations such as dilation and erosion that can be used to expand or contract an image.

2.1.4 Aerial Vehicle Systems

This project is most similar to the system in [23] where they use SLAM for localisation and environment mapping as in [24, 25]. They use a Parrot AR 2.0 drone to capture data about a structure and use the optical metrology system TRITOP to perform deformation measurements on the data. In [26] a system of Micro-Aerial Vehicles (MAVs) is used with on board computer vision capabilities.

2.2 Proposed Solution

Our proposed solution comprises of 4 stages: working with the real drones, training the CNN, processing the CNN output and using ACs.

2.2.1 Drones

We wanted to create a coordinated aerial swarm system to inspect cracks in bridge structures, this is made challenging by environment constraints and mutual interference. The swarm of quadrotors need to perform agile coordination to find goal locations that would correspond to the bridge cracks, meanwhile be preventing collisions between the quadrotors and obstacles in the environment such as the bridge. To address the coordination problem, the Particle Swarm Optimization (PSO) [27] with a centralised strategy will be adapted to coordinate the quadrotors to locate the goal locations. The PSO algorithm optimizes a computational problem by iteratively analysing a population of candidate solutions, represented as particles, and moving these particles around in a search-space. Each particle's movement is influenced by its local best-known position and also the global best-known positions in the searchspace that are updated by all the particles. This simple but effective algorithm is expected to guide the swarm toward the best solutions. However, the PSO algorithm is not appropriate to be directly implemented to aerial swarm robots since the algorithm does not take the collisions of entities (particles) into consideration. To tackle the collision problem and constrain the quadrotors to operate at a safe distance, a force field is proposed to transform PSO into an algorithm, called Force Field Particle Swarm Optimisation (FFPSO), that could be applied to physical robots. This algorithm uses PSO with an adjusted velocity update using an additional force field component to help avoid collisions with the other drones in the swarm. This makes the velocity update retain its simplicity and computational efficiency while the drones explore the space. As the drones have an upward force from flying, they each create a downward air current from their motors. This can be a big issue for the drones as the force is strong enough to direct another drone off its course, flip them upside down and cause crashes. To combat this, the particle vector field of the PSO algorithm is needed to be extended above and below the drone (particle) with a cone shape mimicking the force field. The proposed FFPSO algorithm has been tested and validated in the ARGoS simulator [28] and implemented in a swarm of Crazyflie 2.0s (CF).

Since the CFs were successfully integrated with the FFPSO algorithm then we wanted to use FFPSO with bigger quadrotors such as the Parrot Bebop 2 [29] drones; since they are lightweight, stable and have cameras to record images and video. First testing had to be done with the drones to make sure they were suitable for the algorithm. Testing footage of a bridge was taken manually to help assess the environment of the bridge and get real data on the bridge. Snapshots can be seen in Figure 48 in our Conclusions section.

2.2.2 Convolutional Neural Network

We wanted to use CNNs to be able to classify the data we collected from the bridge so similar sample data was used to train the CNN. The CNN has the structure as in [5], and comes from the Github repository in [6]. They construct their CNN using Python as their programming language with libraries such as Tensorflow and OpenCV. The CNN is trained on data which for this project we used three

datasets: one provided in the Github repository (theirs), one obtained from pavements in Leeds (ours) and one that uses a combination of images from the two former datasets (combined). The data is cut into smaller regions to give not just more training data but so that it can identify a smaller rectangular region as a crack, making classifying more accurate. After it is trained it can be ran using new images as inputs and it produces output images showing the regions of the image where it has identified cracks. It does this by simply taking the class predictions and transforms a copy of the image to only contain the image regions that have the class prediction crack. In all other regions the image is erased and so is left black and we get out an image of just the crack regions. An example of the original images can be seen in Figure 1 with relevant output images that have no post image processing in Figure 2.

The dataset that they used seemed to be not diverse enough to use with our dataset images as the pavement surfaces in our data are more coarse and of different material to the majority of the cracks used in their data. By using some training images from our data as well as their dataset, this gave a more representative output of cracks such as ours. Using the combination of the datasets gave different results shown in a sample of the output images in Figure 4, than just using our data, shown in Figure 3 or just their data shown in Figure 2.

(a) (b)

Figure 1: Original crack images from our dataset. (a) shows the first crack; (b) shows the second crack.

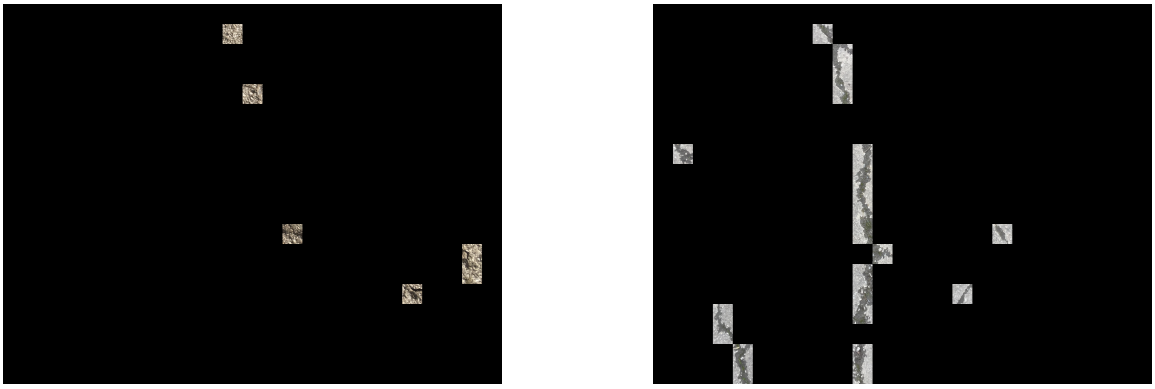
2.3 Morphological Dilation

As previously mentioned work was done on post processing the output images from the CNN, two different filters were used to dilate the images, a 2x2 filter and 3x3 filter, which are filters of the relevant sizes containing all 1s.

The output of the CNN was dilated to open the image out more so that when used with active contours the image had less small enclosed areas, so the resulting active contour is more focused. To use the morphological operation dilation as mentioned in [16], a filter is slid over an image and in our case we use the images predicted class (predicted class value is 0 for crack, 1 for non crack). For each region we have 0 we expand the area according to the filter, it is either expanded to a 2x2 region or a 3x3 region. Figure 5 and 6 show the 2x2 filter and 3x3 filter respectively applied to the original output images Figures 2a and 2b from the CNN trained with their data.

(a) (b)

Figure 2: Output of crack images from the CNN trained using their data. (a) shows the first crack; (b) shows the second crack.



(a) (b)

Figure 3: Output of crack images from the CNN trained using our data. (a) shows the first crack; (b) shows the second crack.

2.4 Active Contours

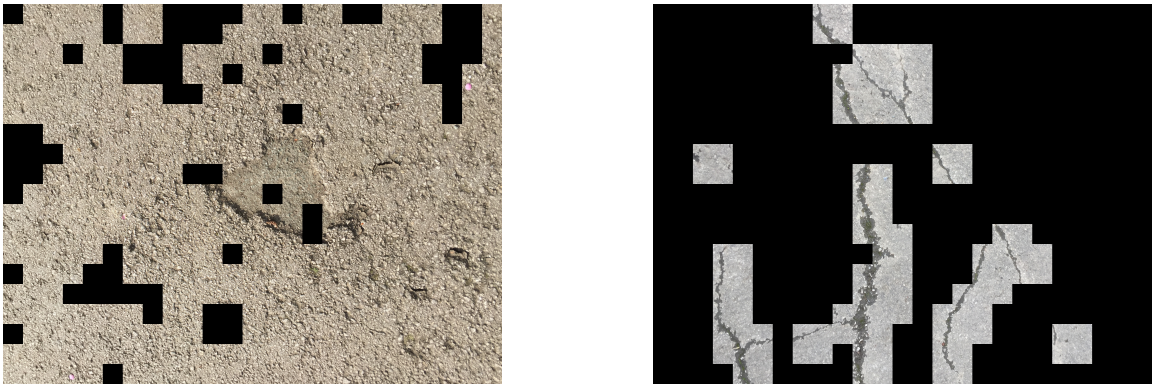
Active contours were used to enhance the visuals of the output images as they highlight any edges or contours in an image, this way the real cracks could be more easily identified in the regions where the CNN identified cracks. The cracks wanted to be more noticeable and to do this active contours known as Morphological Snakes [7, 11, 12, 15, 30, 31]. Using Morphological Snakes on the images such as in Figure 2a and 2b is a good way to define the cracks, but it works better on less noisy images such as the former figure.

The project utilises functions from [17] given about Morphological Snakes, and they are used to apply active contours to the original crack images in Figure 1a and 1b. We use the MorphGAC method that uses Geodesic Active Contours [8] as it is suitable for images with visible contours, wven when the contours might be noisy. However it requires that the image is pre-processed to highlight the contours and recommends using a method that uses an inverted gaussian gradient.

The active contours rely heavily on what values of three parameters are given when using the

(a) (b)

Figure 4: Output of crack images from the CNN trained using the combined data. (a) shows the first crack; (b) shows the second crack.



(a) (b)

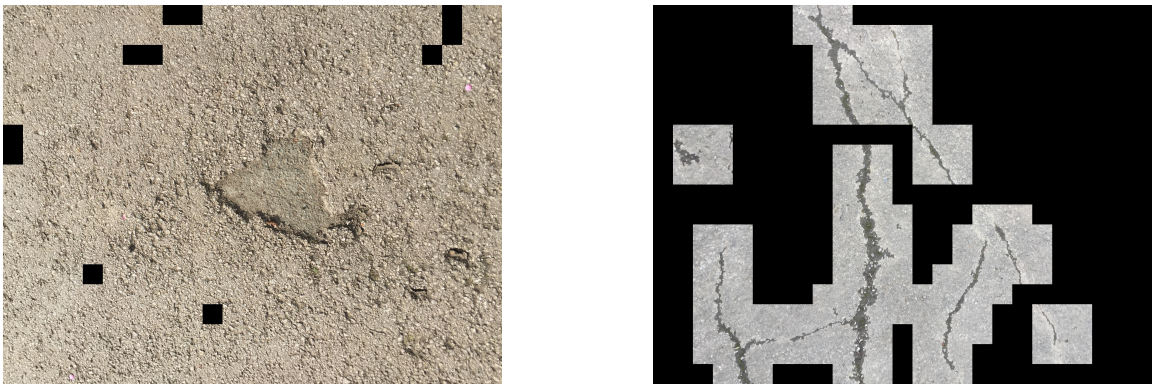
Figure 5: Crack images after using the 2x2 filter. (a) shows the first crack; (b) shows the second crack.

Geodesic Active Contours: standard deviation (sigma), threshold and balloon.

As the images need to be pre-processed we use a form of gaussian filter that uses an inverse gaussian distribution [32, 33] which uses sigma which allows more or less variance around the mean when it is large or small. We use a relatively large sigma as the is a large distribution of values in the images.

The threshold value is closely correlated to the pixel values in the image and their distribution, since it is the threshold to determine where the active contour finds an edge. There are various methods for finding the threshold value, with varying results. The different results of using different thresholding methods can be seen in the experiments for our two images using the different trained CNNs and different initial level sets.

The balloon value determines whether the active contour snakes either contract or expand when evolving; a negative number means they contract while a positive number means that they expand. For our images since we use an initial level set that is outside the cracks then we need a negative number and -1 is commonly used.



(a) (b)

Figure 6: Crack images after using the 3x3 filter. (a) shows the first crack; (b) shows the second crack.

2.5 Impact for a Digital Built Britain

The fast developments in flying unmanned drones and the associated fields of sensors, e.g., cameras, make the inaccessible or difficult-to-access areas (especially the high parts) of bridge structures accessible. Attempts to inspect bridge infrastructure using flying robots have been made in recent years, for example by the European H2020 Industrial Leadership project AEROBI (AErial RObotic System for In-Depth Bridge Inspection by Contact)[1]. Most such attempts have been focusing on improving the functionalities of the aerial robotic systems, for instance, adding a specialised multi-joint arm to a drone. However, performing a complex task like inspecting an entire bridge normally would be not possible for a single flying robot, as the coverage is limited by the battery power the robot carries on board. A further challenge occurs when the space you need to inspect is small or confined, which again could render traditional drones ineffective. A more effective solution would be to use a swarm of small aerial robots that can cooperate with each other to achieve a cohesive set of goals much more effectively. It has been experimentally demonstrated that a swarm of aerial robots working collaboratively will be more efficient and flexible than a single robot working alone [34]. This project was intended to create a coordinated aerial swarm system of Parrot drones to inspect the cracks in bridge structures. Since they are small, agile and have a long flight time they could greatly improve the monitoring coverage and efficiency of surveying a bridge structure.

Significant progress has been made on the digital transformation of the built environment since the 2011 UK Government Construction Strategy, however, this progress has almost entirely been made on new built assets where the digital asset is created at the same time as the physical asset. The capture and utilisation of digital assets for as-built assets significantly lags behind. Yet 70-80% of the buildings and assets the UK will need by 2050 already exist[2].

The CDBB mission around commercial competitiveness and productivity, as well as citizen quality of life and well-being, will be severely compromised unless we can effectively create digital assets of the as-built UK infrastructure. This project is aimed at the development and validation of the prototype of

[1] https://cordis.europa.eu/project/rcn/199152/en

[2] Royal Academy of Engineers, 2010. Engineering a low carbon built environment: The discipline of Building Engineering Physics. https://www.raeng.org.uk/publications/reports/engineering-a-low-carbon-built-environment

an innovative and intelligent aerial swarm robotic system for accurate inspection of reinforced concrete bridges (but is equally applicable to other hazardous or confined spaces where inspection is challenging) featuring a coordination strategy, inspection speed and reliability. It has developed and demonstrated practical insights that will enable the exploitation of new and emerging swarm robotics technologies, and collect and analyse the data of built environments, i.e., bridge structures. As part of this project we intend to create a bridge inspection system without interfering with traffic or endangering human inspectors. Commercialisation of the research is a strong possibility. More importantly, the project targets to monitor the structural integrity of bridges and flag any potential issues, helping to avoid the tragedies like the Morandi bridge collapse[3].

The project is interdisciplinary and generates great impact for a Digital Built Britain, in the aspects including but not limited to:

1. It has explored the social implications of a digital built Britain for health and well-being. The project aims to develop an autonomous system using digital technologies (robotics, AI and data analysis) for monitoring the health of bridges and other structures, which contributes to guarantee the safety and well-being of the people of Britain. The development and dissemination of the project will reassure the public about the health of the bridge infrastructures as well as develop preventative and predictable maintenance strategies.

2. It has explored the implications of a digital built Britain for mobility and transport. Bridges are important links in modern society creating crossings between countries, cities and regions. They lead traffic over rivers, seas or urban streets enhancing mobility and transport, which places great demand on safety, function and operability of bridges. The project creates a bridge inspection system that ensures the safety and structural integrity of bridges. It contributes to avoiding personnel and economic losses due to collapses of bridges lack of inspection and monitoring. The development of autonomous inspection system also does not interfere with traffic, further improving the mobility and transport.

3. It has explored the exploitation of existing or emerging tools, technologies and techniques and their role in delivering a digital built Britain. This project makes use of the emerging digital technologies, i.e., swarm robotics, embedded systems, UAV technologies, Artificial Intelligence algorithms, computer vision techniques, machine learning and deep learning methods, control and system integration, to achieve delivering a feasibility study of the digital built Britain.

4. It has explored ways to leverage data and information to deliver a digital built Britain. The project not only creates on-the-fly processing technologies but also collects real world data of bridge structures, e.g., the data generated on the drones during the inspection of the bridges. To generate greater impact of the research to be conducted in this work and benefit the research community, the dataset and the source code will be made public so that researchers can access the data to investigate the bridge monitoring and assessment.

5. It has explored the commercial challenges and opportunities of a digital built Britain. Bridges have been being built as important links across the world, therefore, there is a big global market

[3] https://www.bbc.co.uk/news/world-europe-48790056

to continuously monitor the structural soundness of these bridges. The developed aerial swarm system has a strong possibility to be commercialised and can be deployed to accomplish such tasks.

In addition, the work conducted in the project will be applied to several areas of interest for a Digital Built Britain:

1. **Society.** The project contributes to the harmony and stability of the society by avoiding personnel and economic losses caused by bridge collapses due to lack of inspection and monitoring, like the one happened to the Morandi bridge.

2. **Digital.** Digital technologies have been developed in this project, i.e., aerial swarm coordination, crack detection and classification, and optimisation algorithms. The state-of-the-art research and technological tools have also be applied to the project, e.g., swarm robotics simulators, UAV platforms, embedded systems, computer vision techniques, machine learning and deep learning methods, control and system integration.

3. **Construction.** The project has developed and demonstrated practical insights into the design, construction, monitoring and maintenance of the city infrastructures, i.e., bridges in this particular case.

4. **Complex integrated systems.** An integrated aerial swarm system has been developed for crack detection of bridge structures. The developed algorithms and methodologies will be beneficial to different applications and the integrated system has a strong possibility to be commercialised.

5. **Data and information.** The real-world data of bridge structures have been collected for the in-depth research of bridge monitoring and assessment. The dataset and the source code will be made public so that researchers can access the data to investigate the bridge monitoring and assessment to generate greater impact.

3 Conclusions

3.1 Expected Outcome/Experimental Results

The expected outcome of analysing the trained CNNs would be that the training set with the combination of data would improve the accuracy of the CNN to find cracks in images. Another prediction would be that in a region where the cracks were not detected very well, dilation of the image output would help the detection of cracks.

Various testing was done for each of the three trained CNNs. For the two images 1 and 2 they were passed into the CNN to classify each image region if it is crack or non-crack, which gives the resulting output images where if a region is classed as non-crack then that section of the image is blacked out. This was done with no post processing to produce the original output images, as well as additional images where the original images were processed using dilation with a 2x2 filter. Both of the images were used with ACs to identify the cracks more precisely so that they could be more easily seen, and the output images would be used as the initial level set to be the outer boundary of the detected cracks.

There are a few tuneable parameters for the AC but the most interesting and focused on one was the threshold value; this is the value used by the AC to determine when it has identified an edge in the image, and so evolution stops. Therefore three different threshold values were evaluated using the three thresholding methods: Minimum, Otsu and Yen which are different depending on the image and were used since they were the best results for the types of images were using. A look at all the thresholding methods the library has can be seen in Figure 7 and 8. Below shows the results with the three trained CNNs.

Both of the improved initial sets in all CNNs show good accuracy to find the edges of the cracks. Image 1 suits thresholds of lower values since it does not contain a crack and so the Minimum and Yen results are best. Dilation only hinders the progress of the AC since there is a limited amount of iterations we can do quickly and so when the image output is dilated we get an outcome similar to when we use the original level set of the images edges. Whereas image 2 does contain a crack so its threshold is best when it is higher as in the Otsu results, so that it stops evolution at the cracks. Dilation helps to connect regions that are part of a continuous crack but are disjointed in the output image, so this way the ACs can find longer portions of the cracks. The AC results ultimately depend on the accuracy of the CNN, if it is not accurate then neither will the ACs.

3.1.1 CNN Trained With Their Dataset

First image 1 was used with ACs using the original image (Figure 2a) as the initial level set.

Then image 1 was used with ACs using the dilated image (Figure 5a) as the initial level set.

The same was done for image 2.

Overall the initial level set for image 1 is very noisy and contains a lot of false positive results for regions of the image, this is why it does not perform well using the AC, as they need very accurate initial level sets. It still performs well for Minimum and yen thresholds and almost reduce the contours to 0. Dilation does not help image one since it only magnifies the false positive results. It works better for image 2 as its shape is similar to the other CNNs.

3.1.2 CNN Trained With Our Dataset

First image 1 was used with ACs using the original image (Figure 3a) as the initial level set.

Then image 1 was used with ACs using the dilated image (Figure 5a) as the initial level set.

The same was done for image 2.

Image 1 shows very little contours for all thresholding methods, even resolving to 0 for Minimum and Yen, which is what was expected since there are no cracks in this image. The dilation made no impact on Minimum and Yen which shows these thresholds are robust. For image 2 the best threshold is the Otsu one as it finds the most of the cracks in the image and within the level sets.

3.1.3 CNN Trained With the Combined Dataset

First image 1 was used with ACs using the original image (Figure 4a) as the initial level set.

Then image 1 was used with ACs using the dilated image (Figure 5a) as the initial level set.

The same was done for image 2.

The output images of the combined dataset were very similar to the results of the dataset using our data, but the result with the combined data has a bit more detail and covers more regions with crack classifications. Therefore the AC results are very similar but here we have more crack definition from using the Yen threshold, and the overall ACs is more connected.

3.1.4 Active Contours

Some small tests were done on the two crack images to show the difference of the initial level set. For all of the tests the smoothness was set to 4, the threshold value is the Otsu threshold of the image and the balloon value is -1.

First the images borders were used as the initial level set, as this can be used for images but is more suited when the edges are well defined and are are easily distinguished from the background. Figure 45 shows the results for both images, the first image shows uniform evolution on most sides of the image as there are no cracks in the image. However there are a few areas where the snakes catch the coarser parts of the pavement. Whereas with the second image we have some of the cracks defined where they are close to the edge, because the snakes find edges here.

In order for active contours to work correctly in a reasonable amount of time, the initial level set needs to be close to the features of interest. To do this the initial level set is initialised as the border of the corresponding output image from the CNN. This works very well for the less noisy image in Figure 1b as seen in Figure 46 but can lead to unsatisfactory results for more noisy images such as in Figure 1a.

Figure 47 shows how the difference of the threshold can give, this value determines when the active contours find a border, the contour evolution will stop in these areas. The values are arbitrary just to illustrate the difference the threshold value has.

3.1.5 Metric Evaluation

After training the CNN using the different datasets their image outputs were evaluated using two different evaluation metrics: Intersection over Union (IoU) and F1-score.

The F1-score is the harmonic average of the precision and recall for binary classification and is a measure of accuracy. To compute the F1-score we use formulas 1, 2 and 3 below:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F_1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

where TP is the number of true positives, FP is the number of false positives and FN is the number of false negatives as shown in Table 1.

	Predicted Class		
Actual Class		1	0
	1	TP	FN
	0	FP	TN

Table 1: Table showing class outcomes comparing the actual class that comes from the ground truth values and the predicted class from the CNN.

The IoU shows the ratio of the overlap and union of two objects, in our case this is the image output given by the CNN and the ground truth values.

These values can depend on what class we care showing to be positive, in the majority case we are looking at the class crack being positive since this is what we want to detect, but in images that do not have cracks this we focus on the non-crack being positive, this way the area of overlap is not 0.

Tables 2 and 3 below show the different F1-score and IoU values of the differently trained CNNs on the two images 1 and 2 and their respective dilated images. Table 2 shows that for image 1 the CNNs original image output has a higher F1-score and IoU value when compared to when it is dilated shown in bold. Highlighted in blue is the highest of the scores in bold for F1-score and IoU, and are both achieved by the CNN trained by the dataset contained by our own dataset, this is probably due to image 1 coming from our whole dataset, but it is worth noting that the value for the combined data is very close to the highest, while the dataset with only their data was much lower. Table 3 shows that for image 2 the CNNs dilated image output has a higher F1-score and IoU value when compared to the original image shown in bold. Highlighted in blue is the highest of the scores in bold for F1-score and IoU, and are both achieved by the CNN trained by the dataset containing only their data, but these values are very close to the values obtained when trained with the combined dataset. Since image 1 does not have a visible crack and image 2 does, then image 1 has to look at the IoU with respect to the non-crack regions, and image 2 looks at the IoU in respect to the crack regions, although image 2 shows a similar pattern in regards to those values.

With these results the CNN trained on the combined dataset has a greater average accuracy, compared to the other trainings, and is not the worst in either image case.

CNN Training Dataset	Image	F1-score	IoU (with respect to)	
			Crack	Non-crack
theirs	1	**0.520**	~~0~~	**0.520**
	1 (dilated)	0.139	~~0~~	0.139
ours	1	**0.987**	~~0~~	**0.987**
	1 (dilated)	0.954	~~0~~	0.954
both	1	**0.966**	~~0~~	**0.966**
	1 (dilated)	0.886	~~0~~	0.886

Table 2: Table showing class outcomes comparing the actual class that comes from the ground truth values and the predicted class from the CNN.

CNN Training Dataset	Image	F1-score	IoU (with respect to)	
			Crack	Non-crack
theirs	2	0.813	0.321	~~0.794~~
	2 (dilated)	**0.859**	**0.559**	~~0.828~~
ours	2	0.771	0.168	~~0.759~~
	2 (dilated)	**0.834**	**0.410**	~~0.812~~
both	2	0.808	0.311	~~0.790~~
	2 (dilated)	**0.851**	**0.533**	~~0.820~~

Table 3: Table showing class outcomes comparing the actual class that comes from the ground truth values and the predicted class from the CNN.

3.1.6 Bridge Data Collecting

A parrot drone was used to take visual footage of a bridge in Liverpool in the UK. Figure 48 shows a view of the whole bridge while Figure 49 shows a close up of the bridges concrete structure.

4 Acknowledgements

This work incorporates results from the research project “Aerial Swarm Robotics for Active Inspection of Bridges” funded by the Centre for Digital Built Britain, under InnovateUK grant number RG96233.

References

[1] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

[2] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” *Biological cybernetics*, vol. 36, no. 4, pp. 193–202, 1980.

[3] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, “Subject independent facial expression recognition with robust face detection using a convolutional neural network,” *Neural Networks*, vol. 16, no. 5-6, pp. 555–559, 2003.

[4] A. Zhang, K. C. Wang, B. Li, E. Yang, X. Dai, Y. Peng, Y. Fei, Y. Liu, J. Q. Li, and C. Chen, “Automated pixel-level pavement crack detection on 3d asphalt surfaces using a deep-learning network,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 10, pp. 805–819, 2017.

[5] Y.-J. Cha, W. Choi, and O. Büyüköztürk, “Deep learning-based crack damage detection using convolutional neural networks,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 32, no. 5, pp. 361–378, 2017.

[6] S. Rajpal, "Crack detection for an autonomous uav." https://github.com/satyenrajpal/Concrete-Crack-Detection, 2019.

[7] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.

[8] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic active contours," *International journal of computer vision*, vol. 22, no. 1, pp. 61–79, 1997.

[9] A. Yezzi, S. Kichenassamy, A. Kumar, P. Olver, and A. Tannenbaum, "A geometric snake model for segmentation of medical imagery," *IEEE Transactions on medical imaging*, vol. 16, no. 2, pp. 199–209, 1997.

[10] M. R. Jahanshahi and S. F. Masri, "Adaptive vision-based crack detection using 3d scene reconstruction for condition assessment of structures," *Automation in Construction*, vol. 22, pp. 567–576, 2012.

[11] P. Andrey and T. Boudier, "Adaptive active contours (snakes) for the segmentation of complex structures in biological images," *Centre de Recherche Public Henri Tudor Copyright Notice*, vol. 181, 2006.

[12] J. H. Giraldo-Zuluaga and A. E. Salazar-Jiménez, "Automatic segmentation of lizard spots using an active contour model," *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 79, pp. 33–40, 2016.

[13] G. Chen, H. Zhang, I. Chen, and W. Yang, "Active contours with thresholding value for image segmentation," in *2010 20th International Conference on Pattern Recognition*, pp. 2266–2269, IEEE, 2010.

[14] G. Li, X. Zhao, K. Du, F. Ru, and Y. Zhang, "Recognition and evaluation of bridge cracks with modified active contour model and greedy search-based support vector machine," *Automation in Construction*, vol. 78, pp. 51 – 61, 2017.

[15] L. Álvarez, L. Baumela, P. Henríquez, and P. Márquez-Neila, "Morphological snakes," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2197–2202, IEEE, 2010.

[16] J. Serra, *Image Analysis and Mathematical Morphology*. Orlando, FL, USA: Academic Press, Inc., 1983.

[17] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[18] K. Thierbach, P.-L. Bazin, W. De Back, F. Gavriilidis, E. Kirilina, C. Jäger, M. Morawski, S. Geyer, N. Weiskopf, and N. Scherf, "Combining deep learning and active contours opens the way to robust, automated analysis of brain cytoarchitectonics," in *International Workshop on Machine Learning in Medical Imaging*, pp. 179–187, Springer, 2018.

[19] C. Rupprecht, E. Huaroc, M. Baust, and N. Navab, "Deep active contours," *arXiv preprint arXiv:1607.05074*, 2016.

[20] D. Marcos, D. Tuia, B. Kellenberger, L. Zhang, M. Bai, R. Liao, and R. Urtasun, "Learning deep structured active contours end-to-end," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8877–8885, 2018.

[21] J. C. Russ, *The image processing handbook*. CRC press, 2016.

[22] J. S. Lim, "Two-dimensional signal and image processing," *Englewood Cliffs, NJ, Prentice Hall, 1990, 710 p.*, 1990.

[23] A. Ellenberg, L. Branco, A. Krick, I. Bartoli, and A. Kontsos, "Use of unmanned aerial vehicle for quantitative infrastructure evaluation," *Journal of Infrastructure Systems*, vol. 21, no. 3, p. 04014054, 2014.

[24] E. Michaelsen and J. Meidow, "Stochastic reasoning for structural pattern recognition: An example from image-based uav navigation," *Pattern Recognition*, vol. 47, no. 8, pp. 2732–2744, 2014.

[25] J. Park, S. Im, K.-H. Lee, and J.-O. Lee, "Vision-based slam system for small uavs in gps-denied environments," *Journal of Aerospace Engineering*, vol. 25, no. 4, pp. 519–529, 2011.

[26] L. Meier, P. Tanskanen, L. Heng, G. H. Lee, F. Fraundorfer, and M. Pollefeys, "Pixhawk: A micro aerial vehicle design for autonomous flight using onboard computer vision," *Autonomous Robots*, vol. 33, no. 1-2, pp. 21–39, 2012.

[27] J. Kennedy, "Particle swarm optimization," *Encyclopedia of machine learning*, pp. 760–766, 2010.

[28] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, *et al.*, "Argos: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm intelligence*, vol. 6, no. 4, pp. 271–295, 2012.

[29] S. Parrot, "Parrot bebop 2," *Retrieved from Parrot.com: http://www.parrot.com/products/bebop2*, 2016.

[30] A. Hoogi, A. Subramaniam, R. Veerapaneni, and D. L. Rubin, "Adaptive estimation of active contour parameters using convolutional neural networks and texture analysis," *IEEE transactions on medical imaging*, vol. 36, no. 3, pp. 781–791, 2016.

[31] M. Gavilán, D. Balcones, O. Marcos, D. F. Llorca, M. A. Sotelo, I. Parra, M. Ocaña, P. Aliseda, P. Yarza, and A. Amírola, "Adaptive road crack detection system by pavement classification," *Sensors*, vol. 11, no. 10, pp. 9628–9657, 2011.

[32] E. Schrödinger, "Zur theorie der fall-und steigversuche an teilchen mit brownscher bewegung," *Physikalische Zeitschrift*, vol. 16, pp. 289–295, 1915.

[33] M. von Smoluchowski, "Notiz uiber die berechnung der brownschen molekularbewegung bei der ehrenhaft-millikanschen versuchsanordning," *Phys. Z*, vol. 16, pp. 318–321, 1915.

[34] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, “A survey on aerial swarm robotics,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 837–855, 2018.

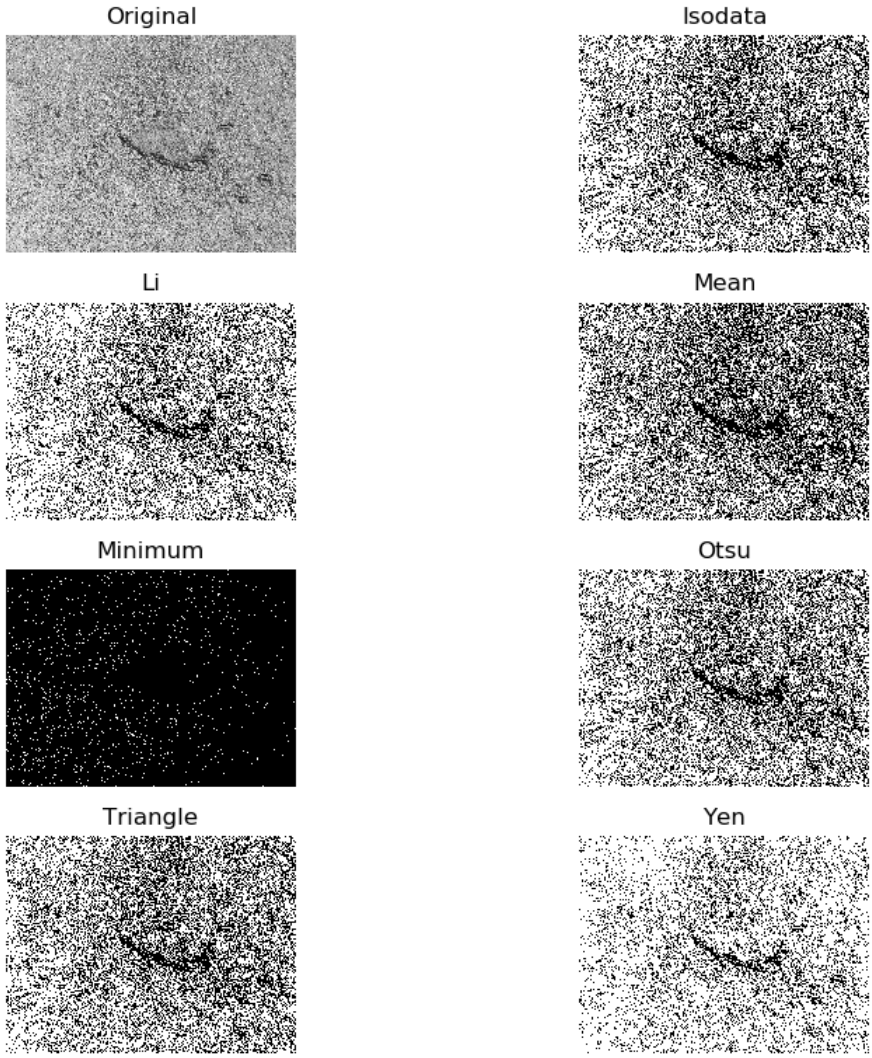


Figure 7: The different image outputs for using different thresholding methods used by the skimage library on image 1.

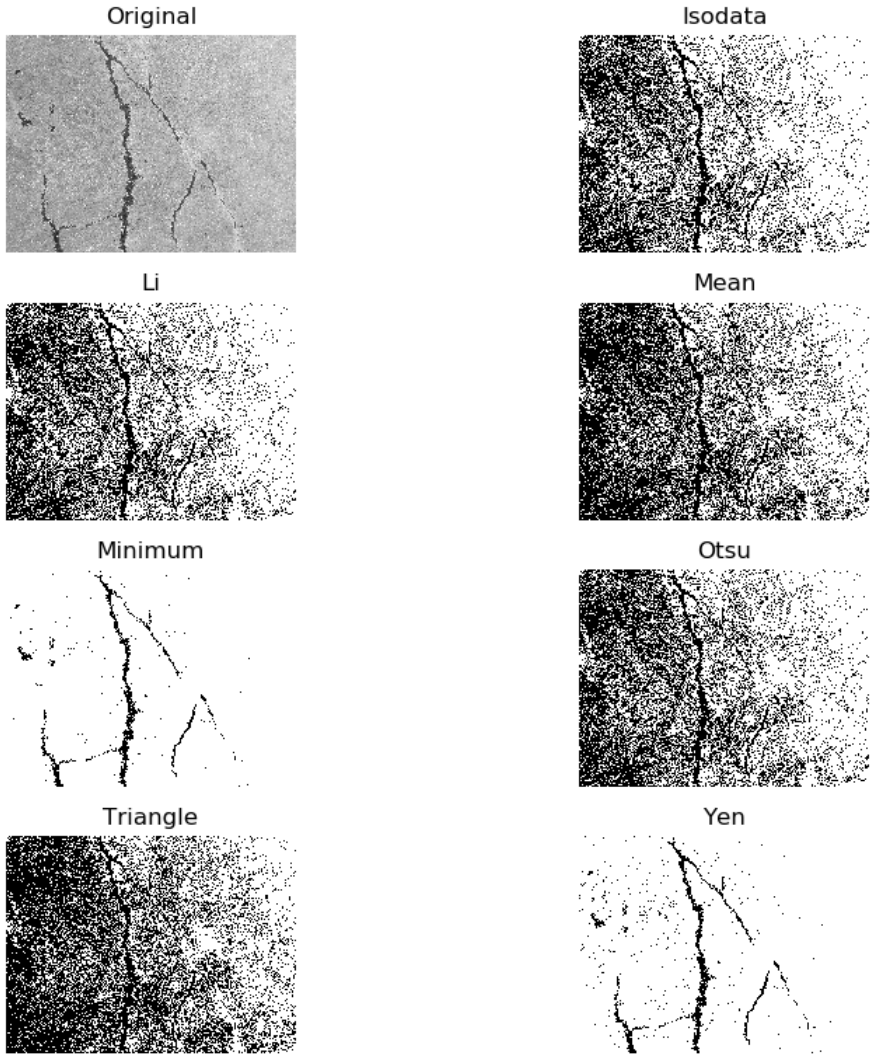


Figure 8: The different image outputs for using different thresholding methods used by the skimage library on image 2.

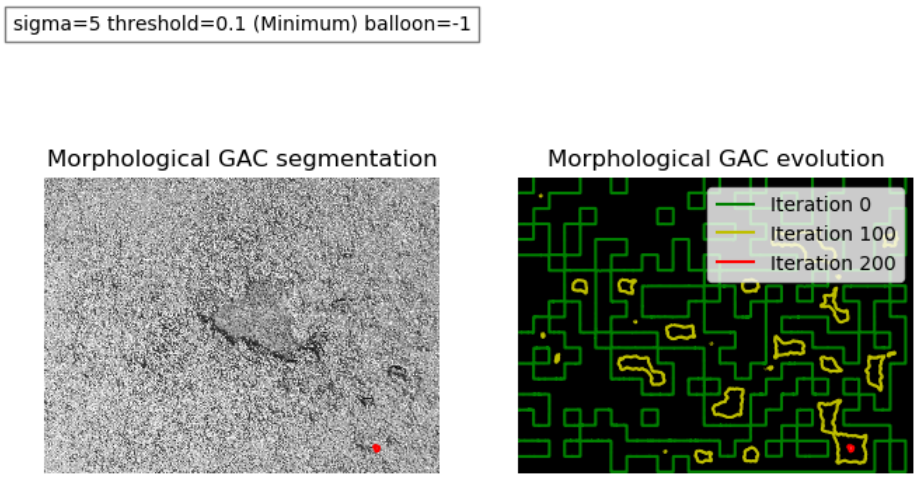


Figure 9: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Minimum method.

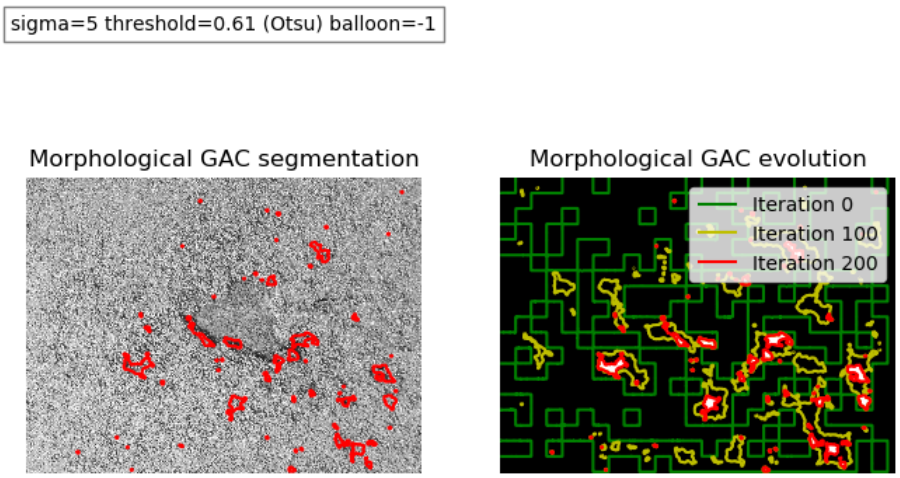


Figure 10: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Otsu method.

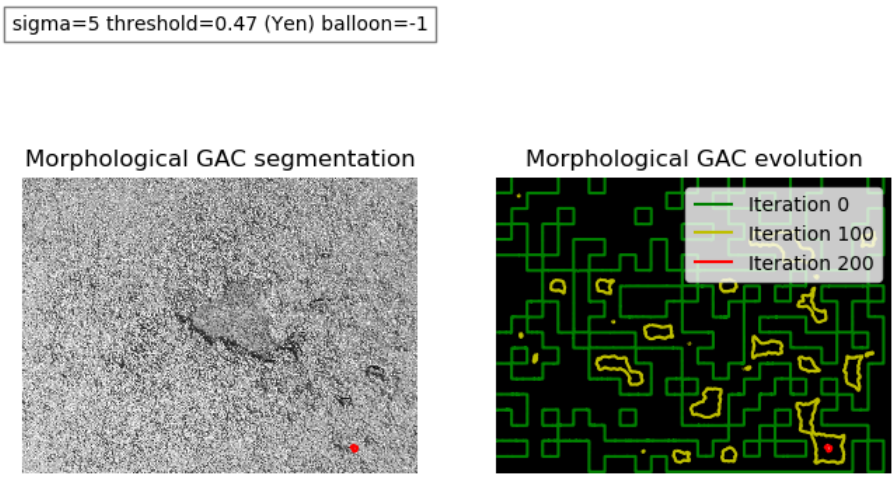


Figure 11: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Yen method.

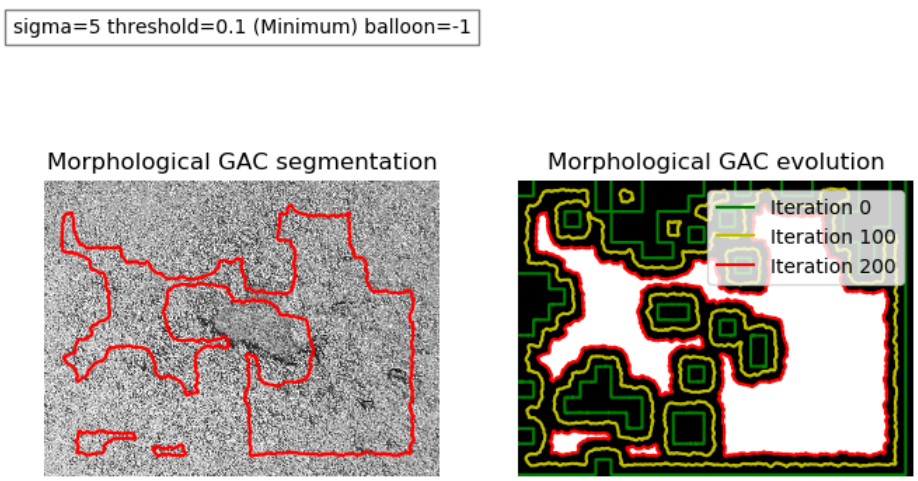


Figure 12: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Minimum method.

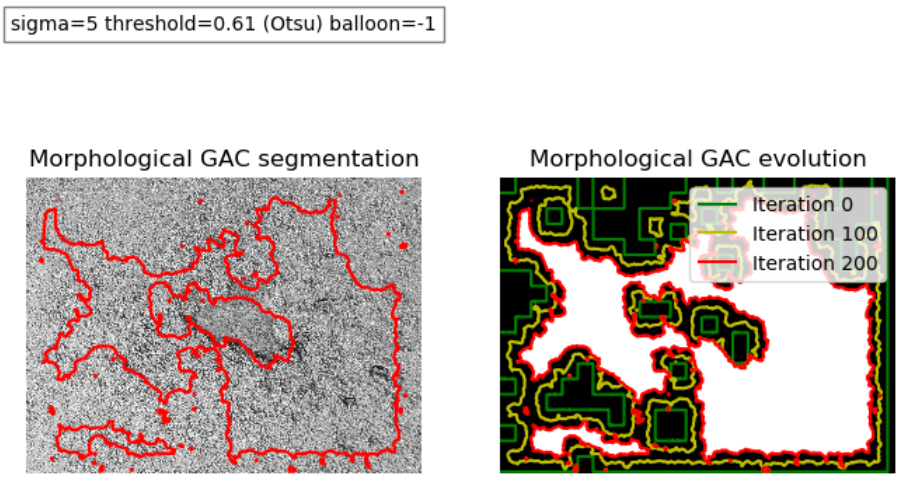


Figure 13: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Otsu method.

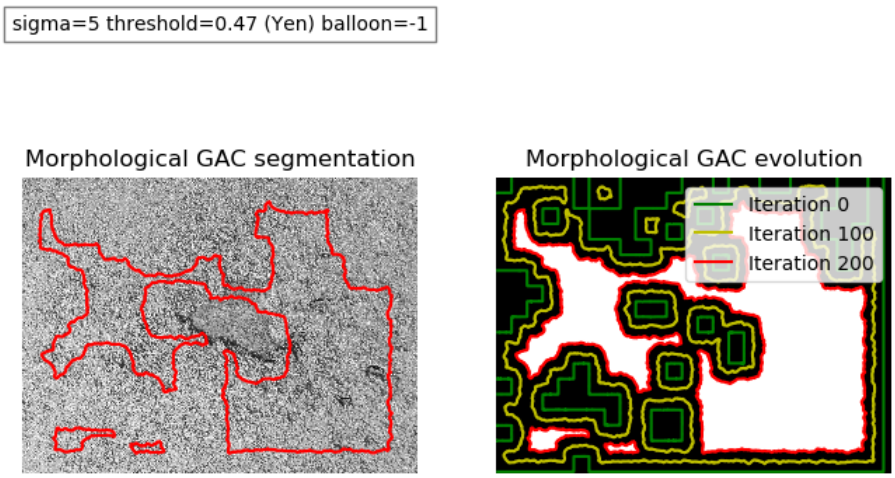


Figure 14: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Yen method.

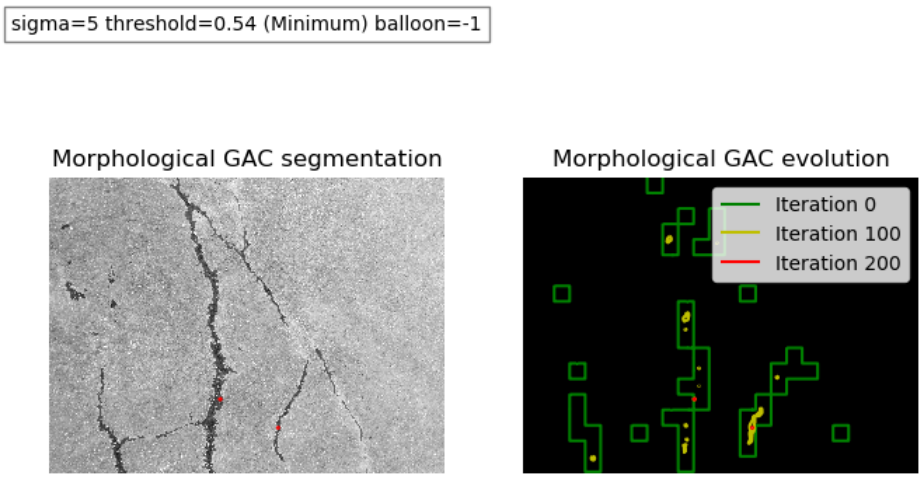


Figure 15: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Minimum method.

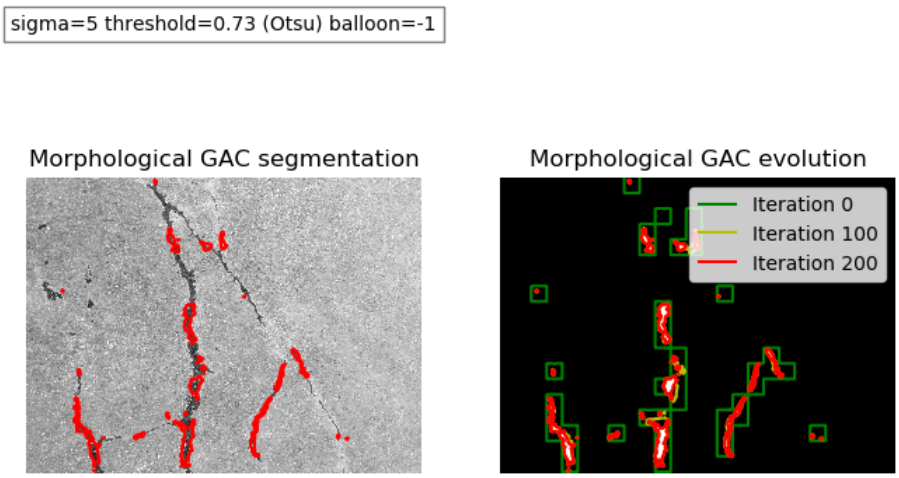


Figure 16: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Otsu method.

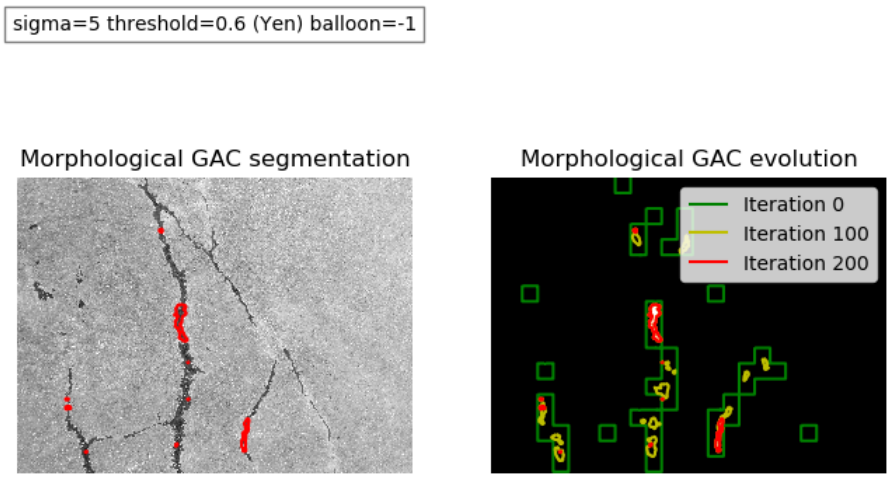


Figure 17: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Yen method.

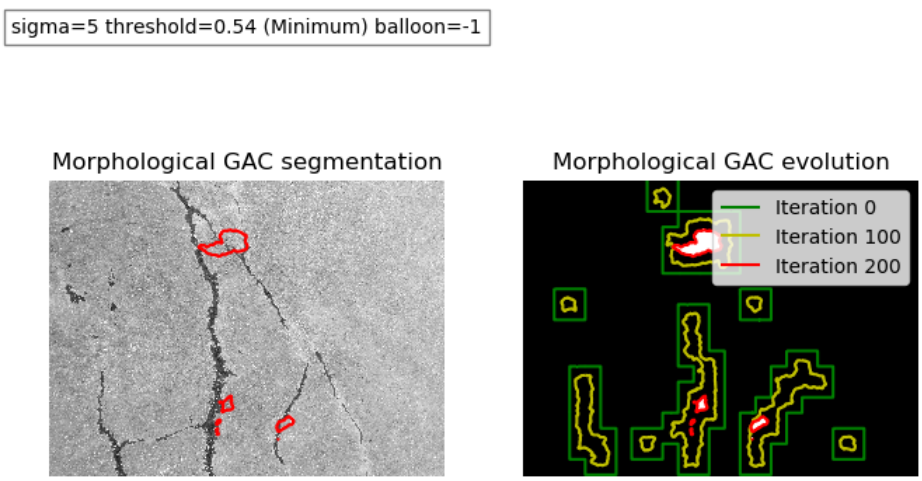


Figure 18: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Minimum method.

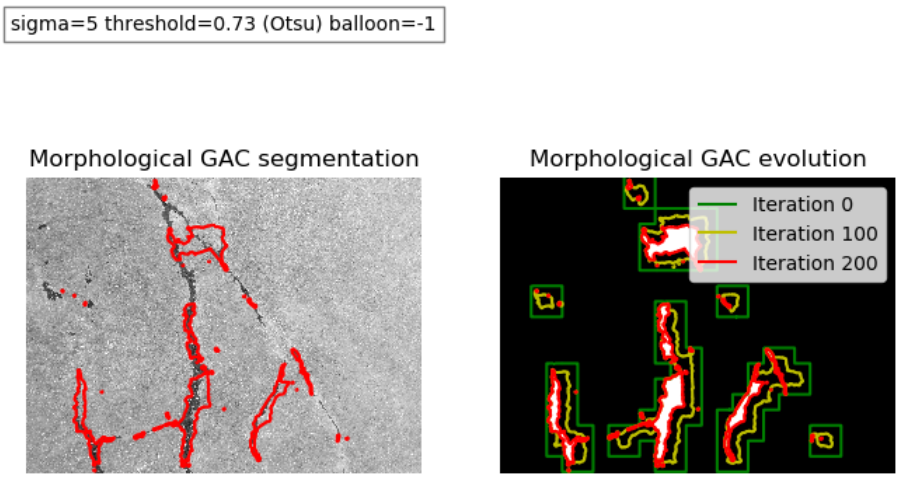


Figure 19: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Otsu method.

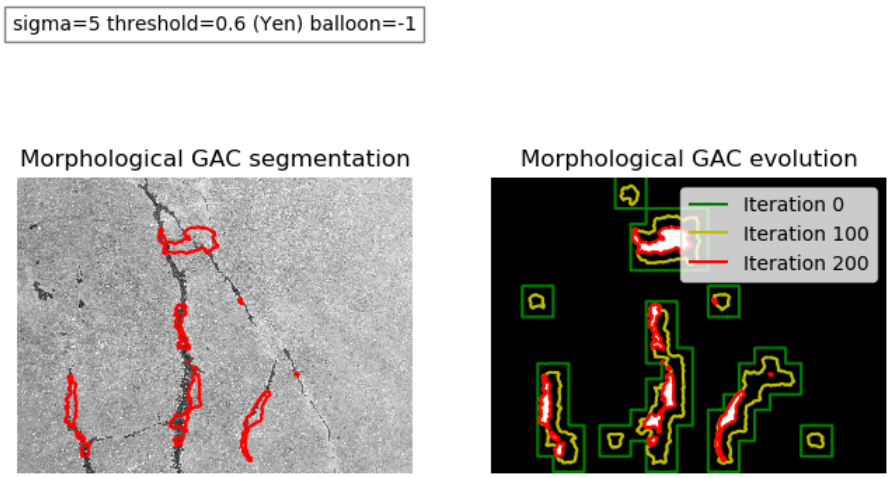


Figure 20: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Yen method.

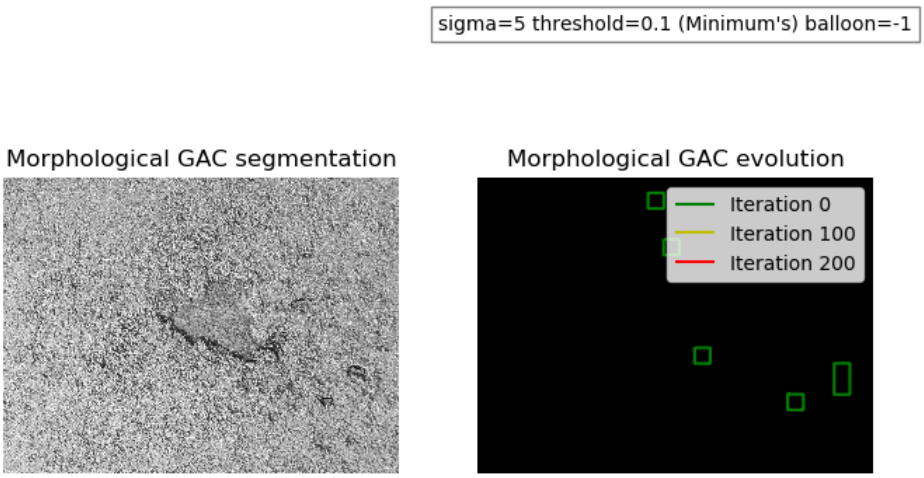


Figure 21: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Minimum method.

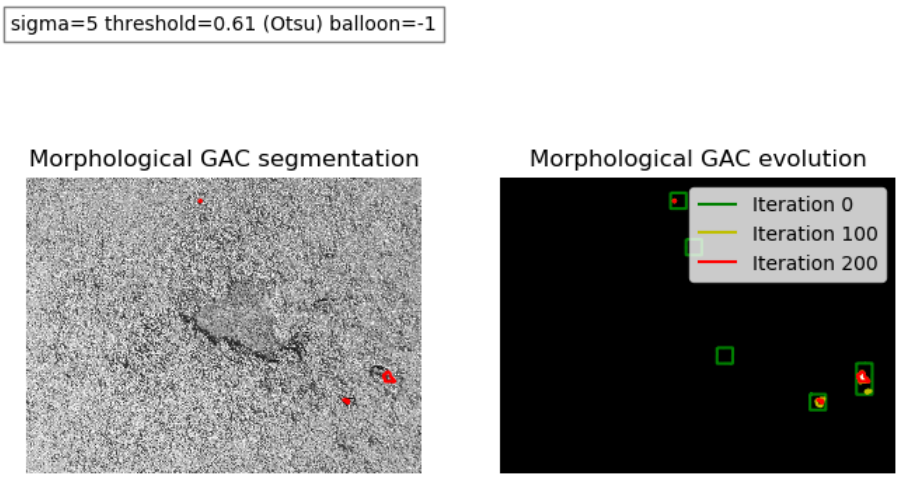


Figure 22: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Otsu method.

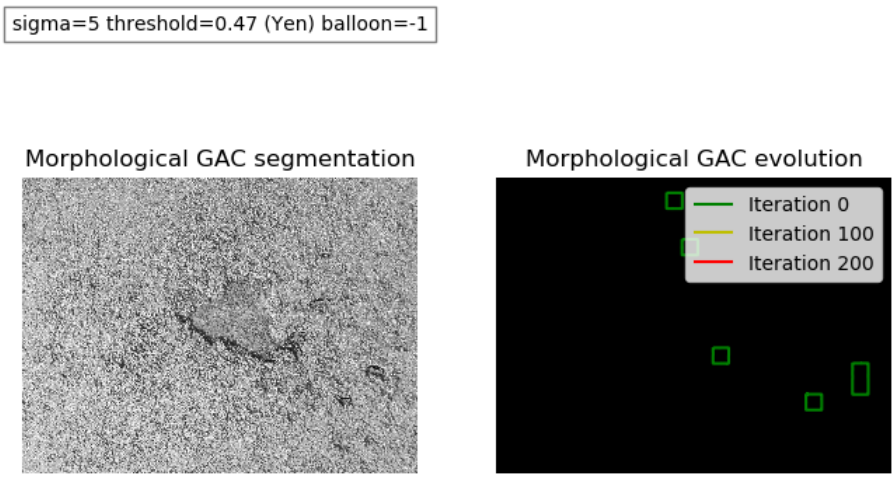


Figure 23: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Yen method.

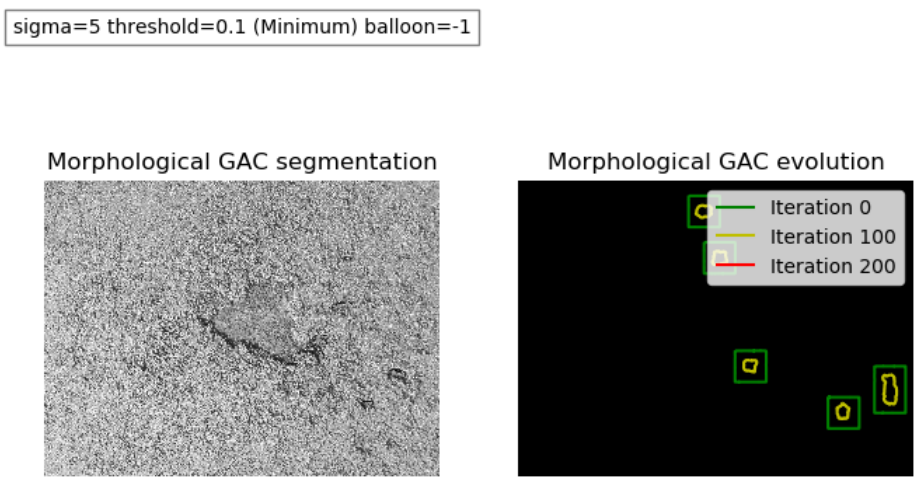


Figure 24: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Minimum method.

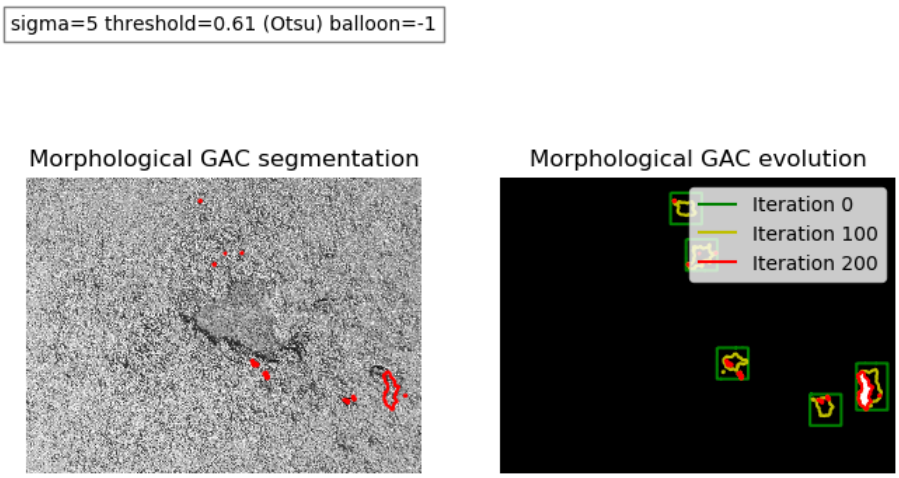


Figure 25: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Otsu method.

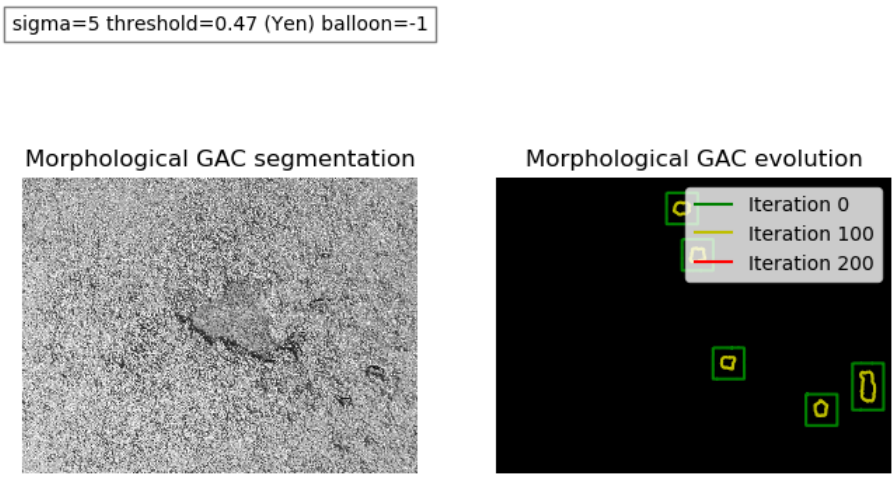


Figure 26: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Yen method.

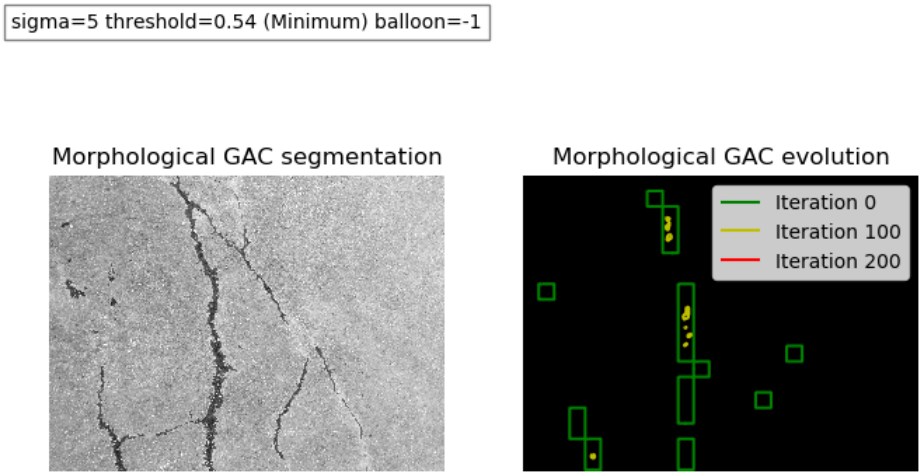


Figure 27: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Minimum method.

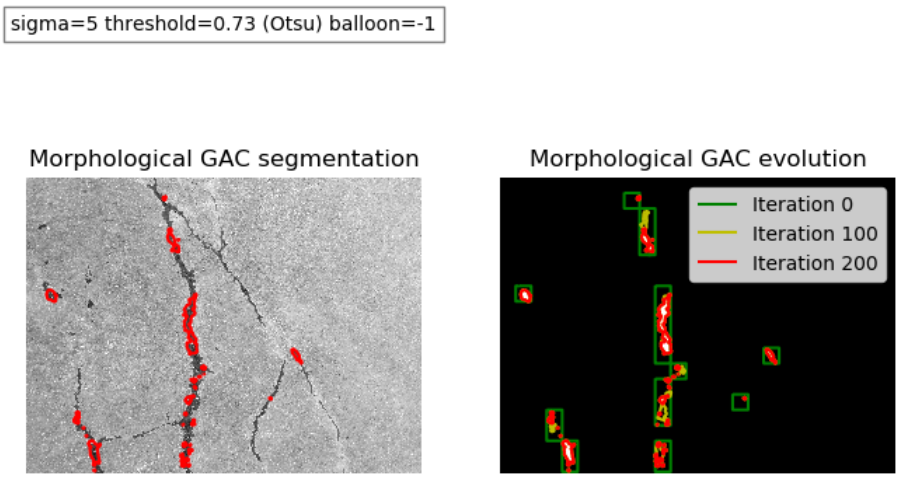


Figure 28: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Otsu method.

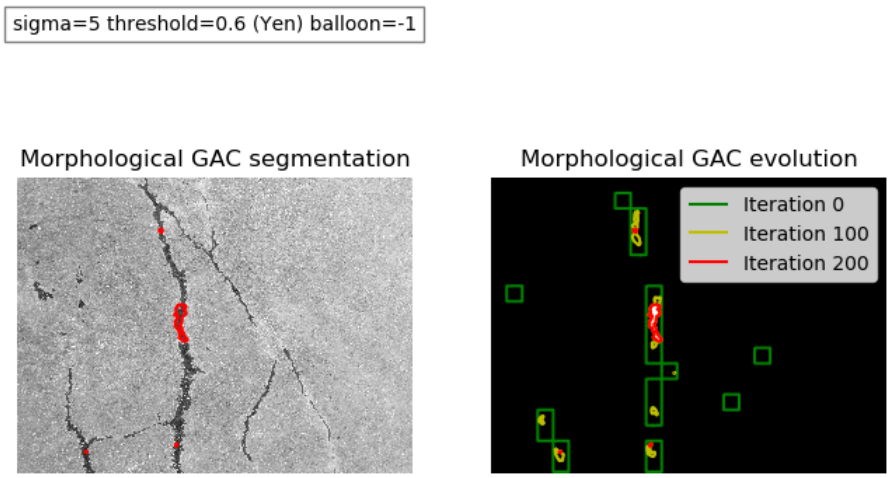


Figure 29: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Yen method.

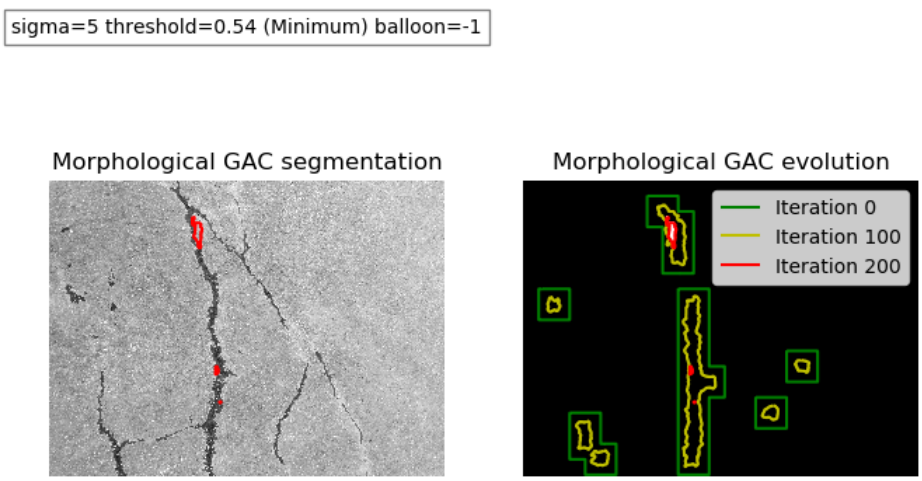


Figure 30: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Minimum method.

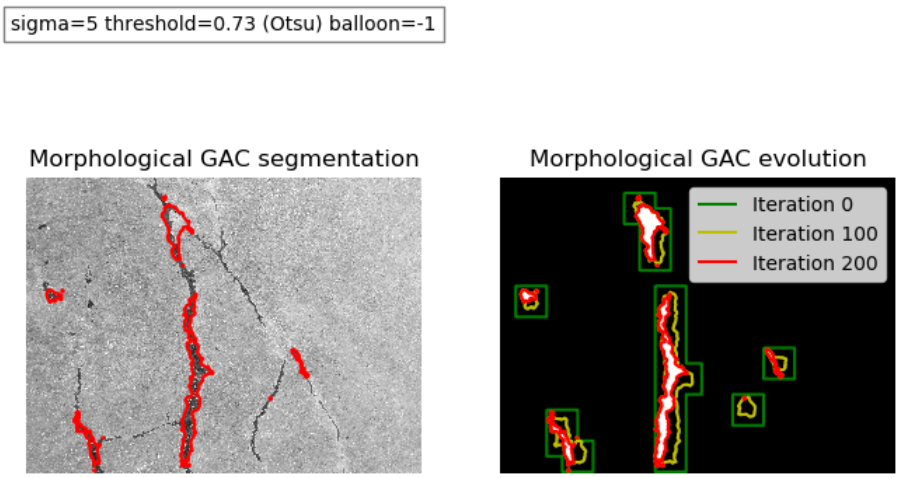


Figure 31: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Otsu method.

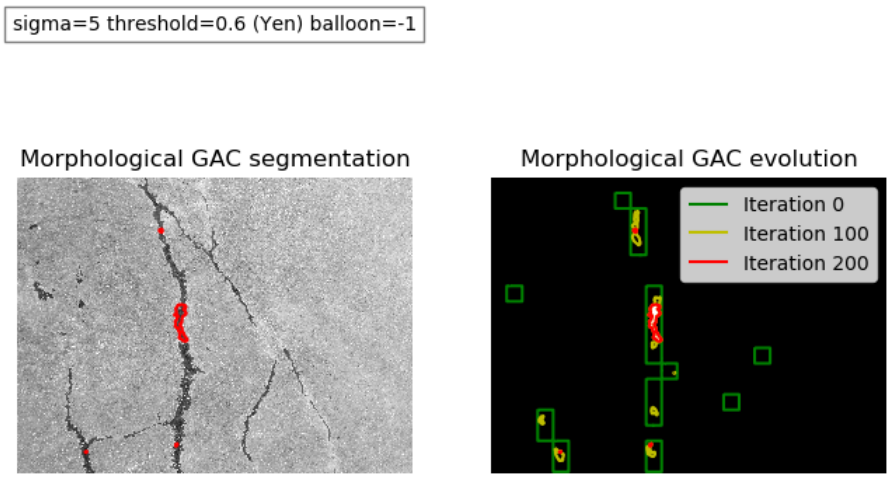


Figure 32: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Yen method.

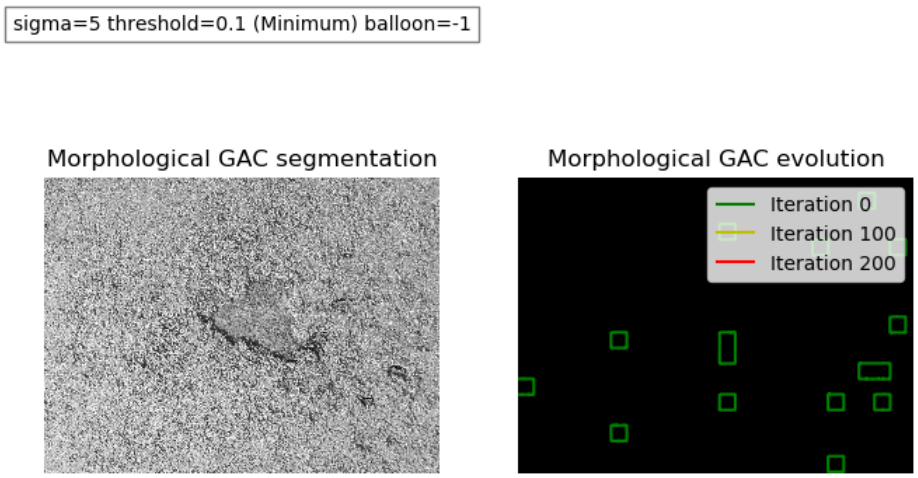


Figure 33: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Minimum method.

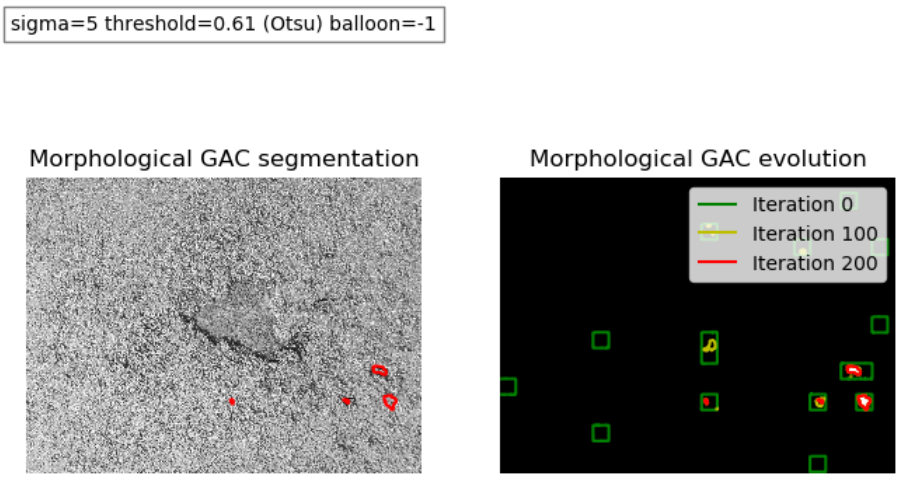


Figure 34: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Otsu method.

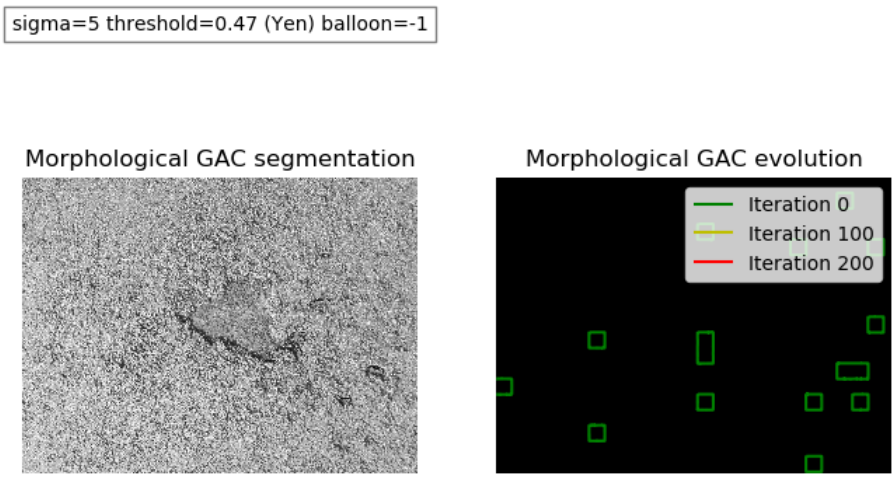


Figure 35: Image 1 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Yen method.

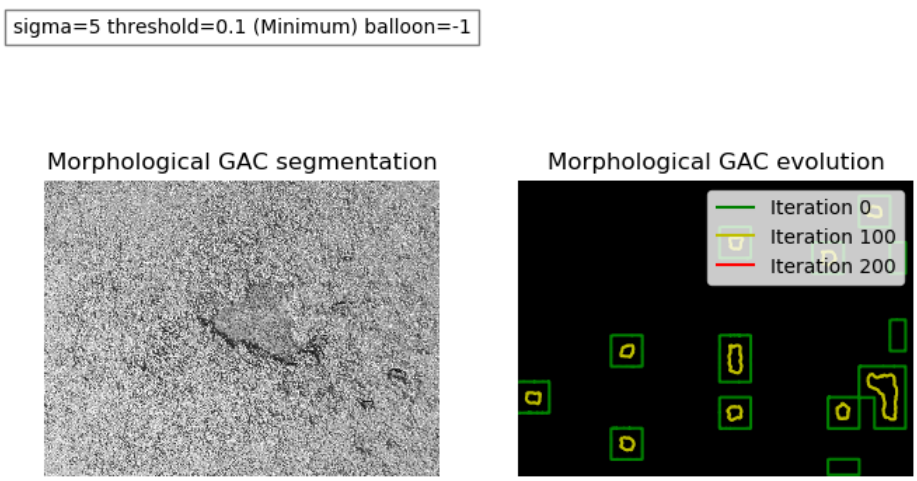


Figure 36: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Minimum method.

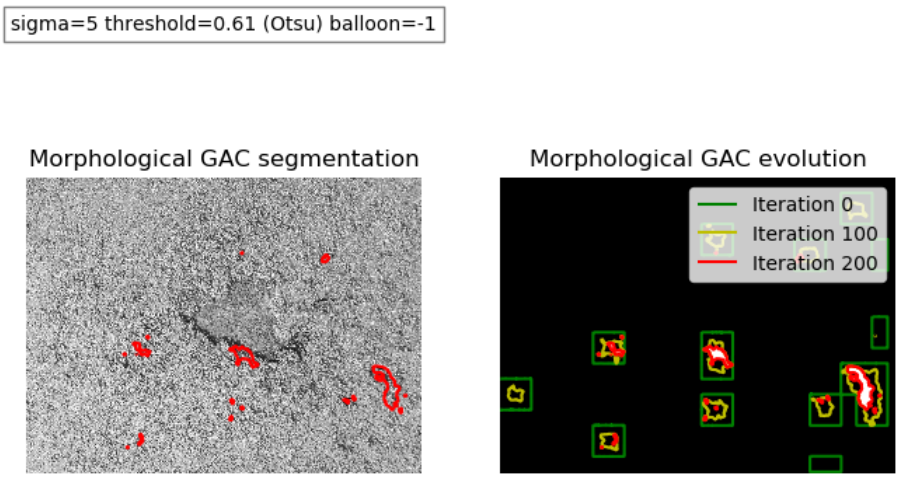


Figure 37: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Otsu method.

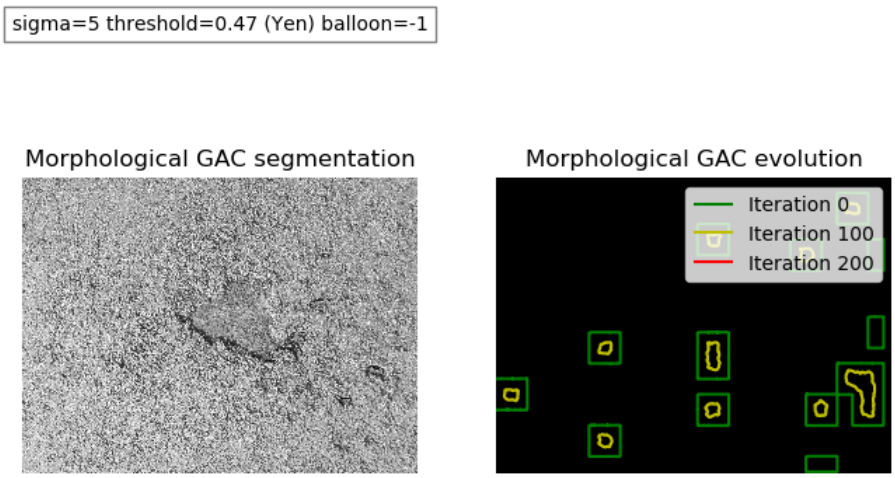


Figure 38: Image 1 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Yen method.

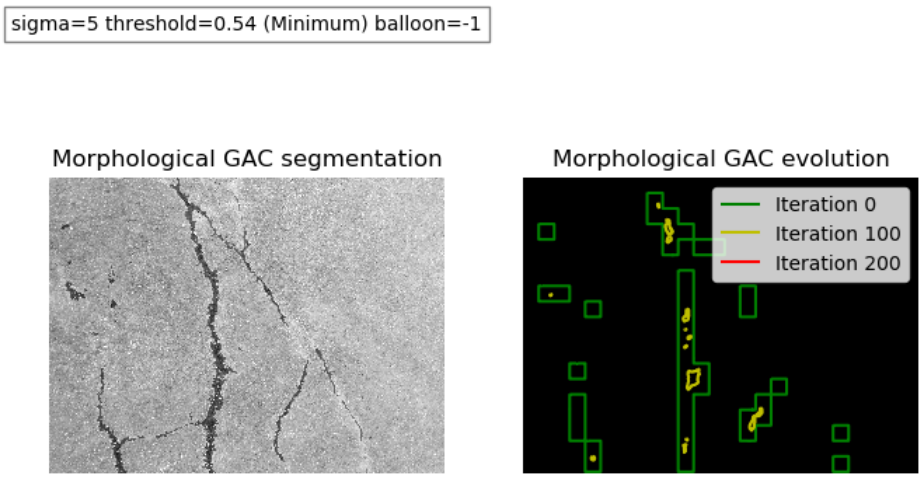


Figure 39: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Minimum method.

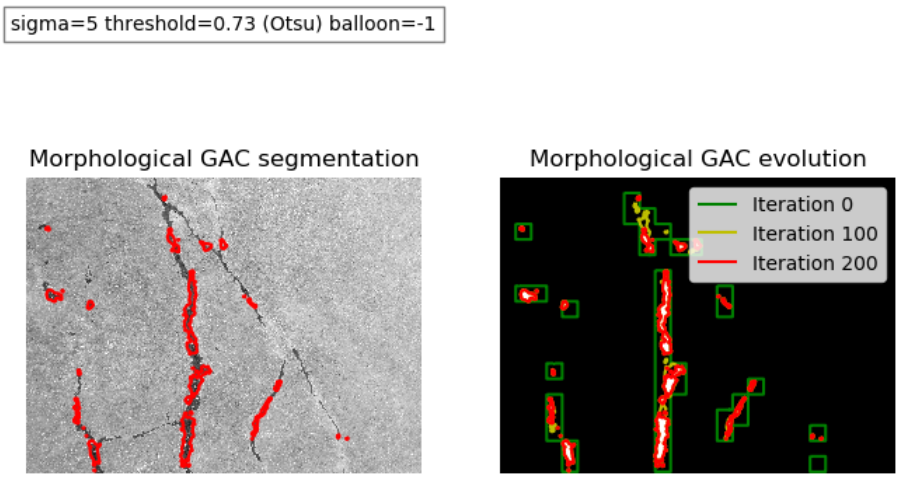


Figure 40: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Otsu method.

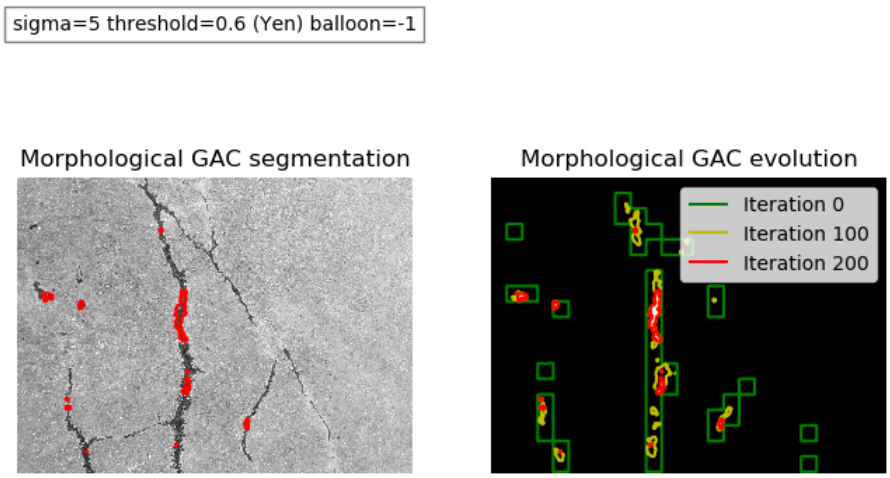


Figure 41: Image 2 with AC acting on it with the initial level set shown by the green line and with a threshold value using the Yen method.

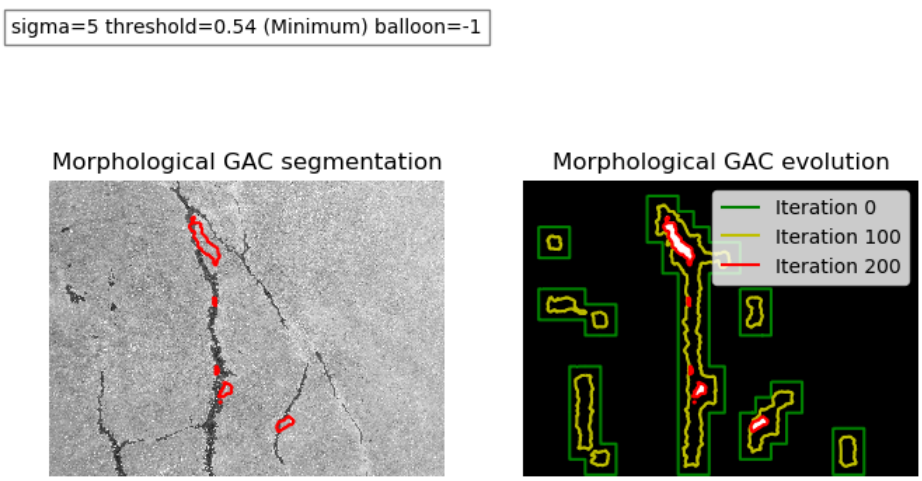


Figure 42: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Minimum method.

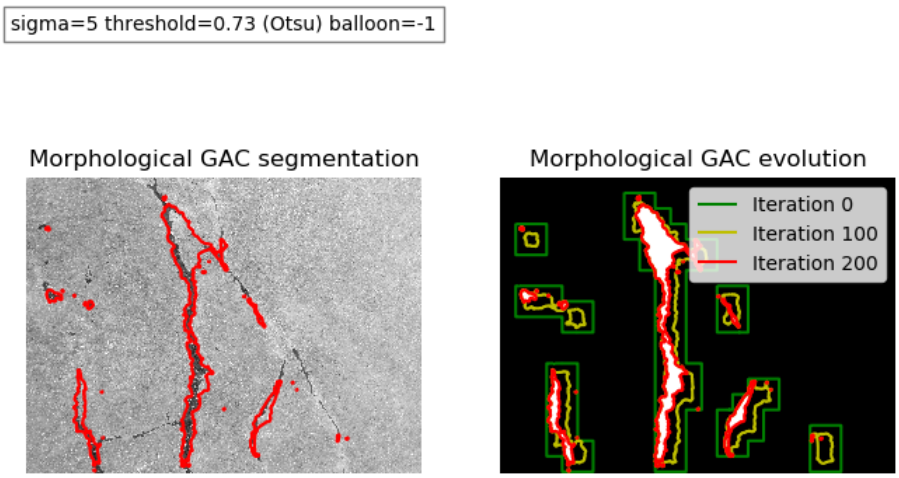


Figure 43: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Otsu method.

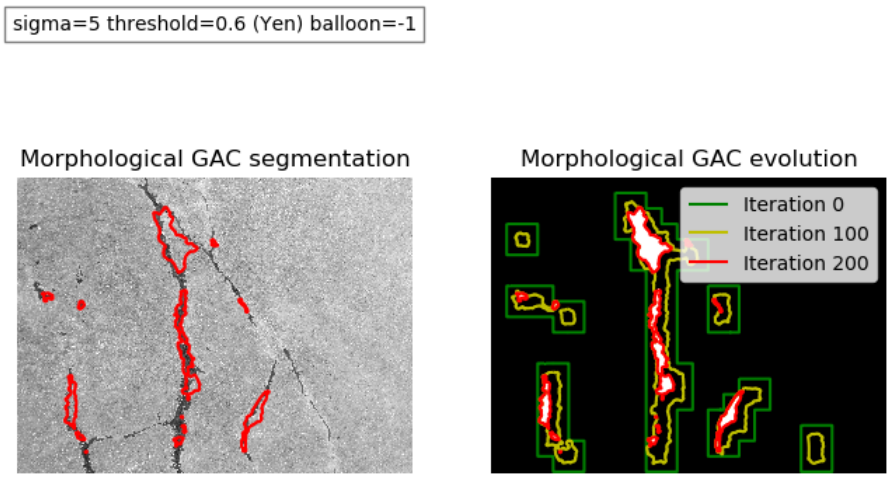


Figure 44: Image 2 with AC acting on it with the dilated image as the initial level set shown by the green line and with a threshold value using the Yen method.

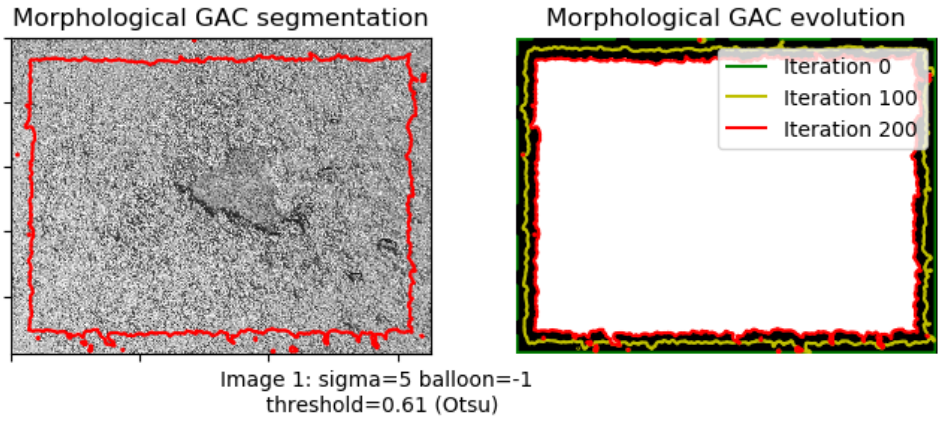


Figure 45: Morphological Snakes on the two original crack images using the original initial level set.

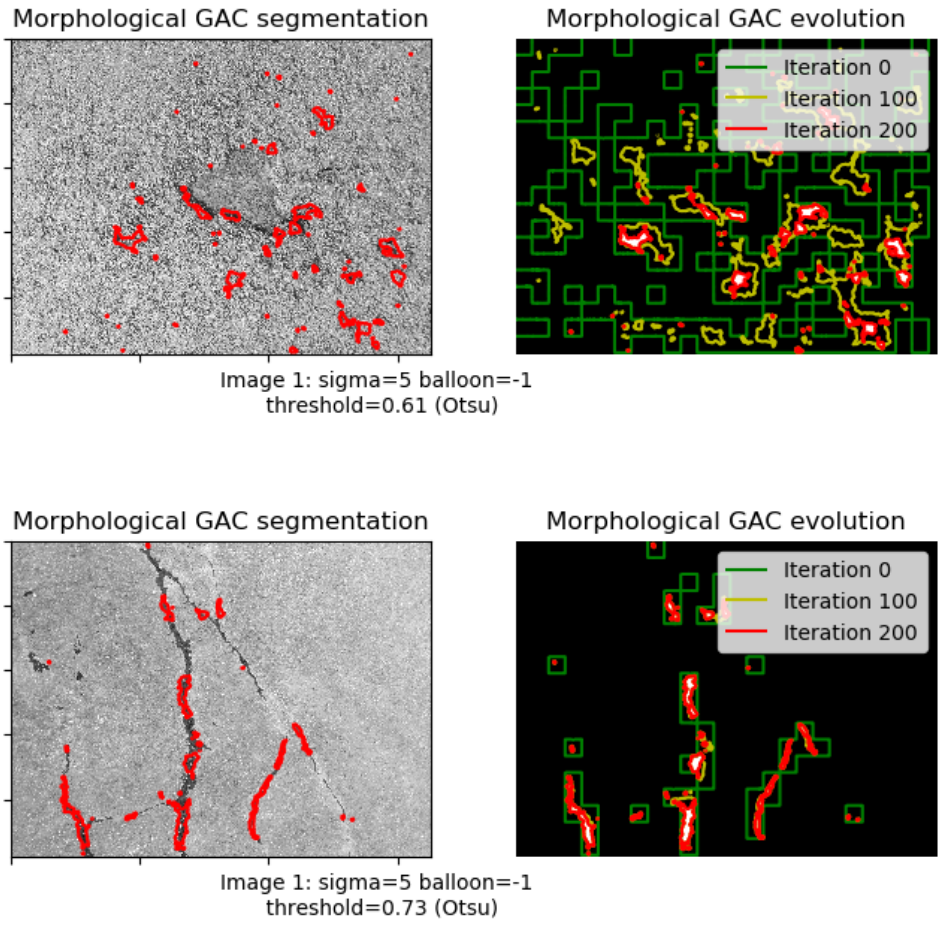


Figure 46: Morphological Snakes on the two original crack images using the initial level set as the output image obtained by the CNN.

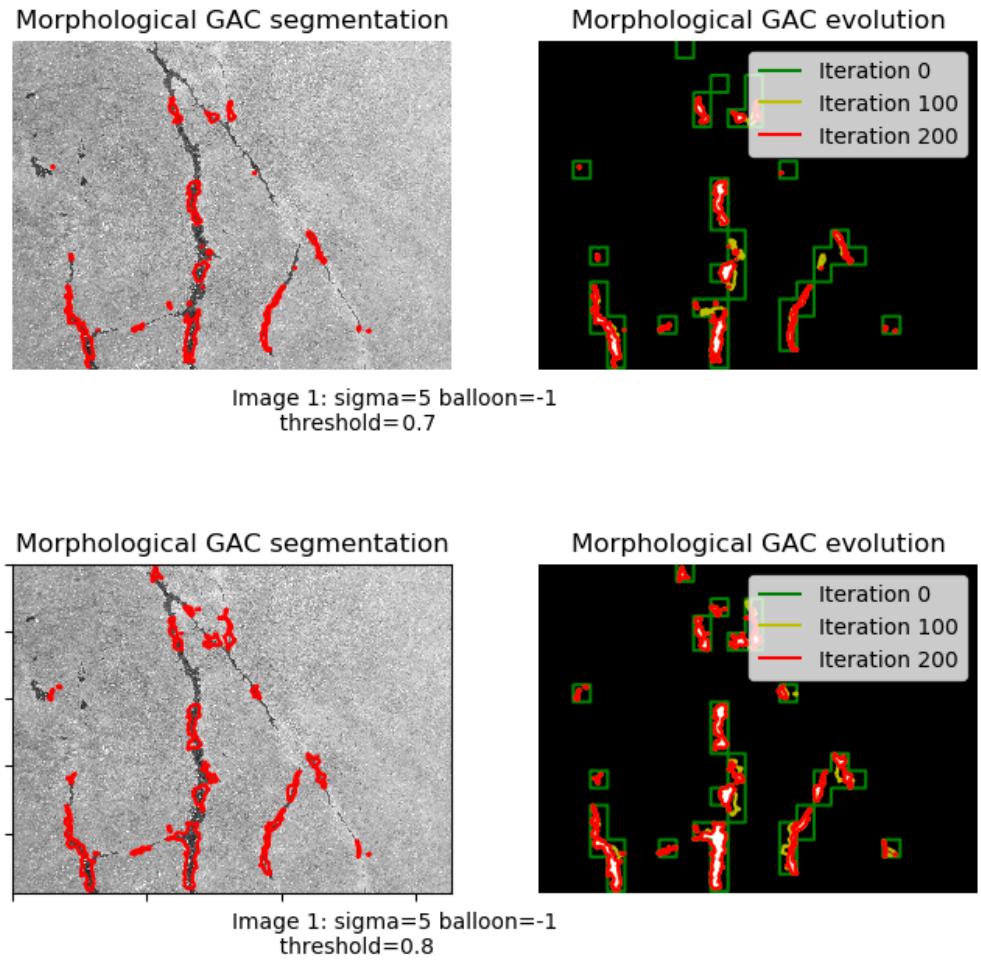


Figure 47: Morphological Snakes on the second crack image using the initial level set as the output image obtained by the CNN and using a threshold value of 0.7 and 0.8 respectively.

Figure 48: Snapshot of a real bridge taken by drone footage.

Figure 49: Snapshot of a real bridge, showing the concrete structure.