SUNSPOT DETECTION AND TRACKING USING IMAGE PROCESSING TECHNIQUES

Ruben Du Toit, Günther Richard Drevin and Roelf Du Toit Strauss
North-West University, Potchefstroom, South Africa

ABSTRACT

Space weather forecasting has become increasingly important over the past five decades as we have become more dependent on technology. Systems such as wireless communication, railway signaling, high voltage power grids and satellites can all be interrupted by solar activity events.
The detection and tracking of sunspots as well as the prediction of solar eruptions forms a part of the attempt to predict space weather. To make predictions regarding the behaviour of a sunspot it is necessary to track and monitor its evolution over time.
In this paper the use of Image Processing techniques to detect and track sunspots will be investigated.

KEYWORDS

Sunspots, Solar Activity, Detection, Tracking, Space Weather

1. INTRODUCTION

Sunspots are dark areas located on the outermost layer of the Sun that emits the light we see from Earth (Curto et al., 2008). By studying the morphology of sunspots, it may be possible to forecast space weather events. Since the temperature of a sunspot is lower than the area surrounding the sunspot, it appears darker and therefore becomes visible. Small dark spots, called pores, form on the photosphere and either decay early in their lifecycle or develop into sunspots. As sunspots start to enlarge, small portions tend to break away from the original sunspot. When a sunspot becomes large enough it forms a double-ended group with opposite magnetic polarities on either side (Curto et al., 2008).

Areas where sunspots are located on are known as active regions, which often lead to solar flares and coronal mass ejections (CMEs). When significant changes at the photospere of the Sun occur, it implies that the overlying magnetic field that is undergoing rearrangement must be strong. This is presumably in the low corona and the energy transported by magnetic disturbances propagating through the chromosphere to the photosphere may be an important component in the flare chromosphere energization (Fletcher et al., 2011) Solar flares occur after the magnetic field topology is rearranged and the magnetic field lines start reconnecting, resulting in solar flares (Guarcello et al., 2019). When these flares occur, large amounts of energy that was stored in the magnetic field is released. These energies are deposited both locally at the point of reconnection and onto the underlying dense chromosphere at the loop footprints of the field lines that the electron beams move along. Solar flares are often followed by another solar event known as a CME. When solar flares occur at active regions, forms of radiation, such as X-rays and radio waves take about eight minutes to reach the Earth from the Sun. Stray X-rays can disrupt the ionosphere enough to disrupt radio communications. They are not usually dangerous to the Earth, rather they act as indicators of potential coronal mass ejections (Guarcello et al., 2019).

Coronal mass ejections are large eruptions of magnetic fields and plasma that are launched into the helio- sphere. These eruptions carry energized electrons which produce various sources of high-intensity plasma emissions (Carley et al., 2017). These storms can affect and damage not only satellites that provide wireless communication from space, but also ground based systems, such as railway signalling, high voltage power transmission grids and telecommunications cables (Colak and Qahwaji, 2007). These CMEs can take a few hours to a few days to reach the Earth (Colak and Qahwaji, 2007).

Communication and power systems are critical for modern society to function optimally. However, these systems are often interrupted by natural events, which are beyond the control of humankind (Colak and Qahwaji, 2007). An example of these natural events is solar activity. Space weather research has taken a major role in the prediction of solar activity, especially since these events are the cause of the interruptions of wireless communications systems, railway signalling, high voltage power transmission grids, telecommunications cables and satellites (Colak and Qahwaji, 2007). Space weather refers to the time-variable conditions in the space environment that could in the worst-case scenario, endanger human life or health by affecting ground-based or space-borne technological systems, according to Koskinen et al. (2001). Being aware of and possibly avoiding the consequences of space weather events are the most important social and economic aspects of space weather. This can be done by efficient warning and detection systems that allow for preventive measures to be taken (Koskinen et al., 2001). Several attempts at systems that can detect sunspots exist (Colak and Qahwaji, 2007). In this section, the background on important concepts and existing systems will be discussed to give insight into this study.

The primary aim of this study is to develop and test various image processing techniques for the detection and tracking of sunspots. The algorithms will be implemented in the programming language Python. Due to the slow rate which sunspots change and move, accuracy in each algorithm is the primary aim instead of how fast it can be executed.

This paper is based on the Master's dissertation of the first author (Du Toit, 2022).

Previous work done on the detection and tracking of sunspots include a system that uses automated detection, characterization and tracking of sunspots on continuum images done in a study by Goel and Mathew (2014). A second approach is an automated sunspot detection using morphological reconstruction and adaptive region growing techniques done by Yu et al. (2014). The last is a system that automatically detect sunspot activities using an advanced detection mode done by Manish et al. (2014). Most previous work made use of different data sets (time intervals) and mainly focussed on determining the number of sunspots on each image, comparing this with available sunspot numbers.

In this paper however, the first aim will be to identify the sunspots and verify the results visually. The main aim of the paper is the tracking of sunspots over a number of days. Once a given sunspot can be tracked its evolution over time can also be studied.

The different methods investigated will be discussed in Section 2 and the results obtained with these methods will be elaborated on in Section 3. Finally conclusions and further work will be discussed in Section 4.

2. METHOD

The data set that was used consists of 20 different dates of images in 4096x4096 resolution with some taken on the same day, but a few hours apart. Four images were taken in 2011, three images in 2012, six images in 2013, six images in 2014 and one image in 2015 that was used for implementing the algorithms on. The data set was downloaded from the National Aeronautics and Space Administration (NASA) website[1].

For the purpose of identifying sunspots, three object detection methods: Canny edge detection; template matching; and watershed segmentation were implemented and tested using images of the Sun. Finally, a novel object tracking algorithm for tracking sunspots was developed and tested on the same images.

2.1 Canny Edge Detection

A high-level overview of the steps for identifying sunspots by using the Canny algorithm is:

1. Identification of the sunspots by using the Canny algorithm of the OpenCV library to detect all the edges of sunspots on the images;
2. The merging of all the bounding boxes that were created from the detected edges. All the bounding boxes are tested to see if they are within a certain distance of pixels of each other. If the boxes are within the distance, a new large single bounding box is created that replaces all the small boxes.

[1] SDO — Solar Dynamics Observatory. https://sdo.gsfc.nasa.gov/

The new bounding box is created with the smallest top left coordinates and the largest bottom right coordinates. The top right corner is the ending x-coordinate and the starting y-coordinate. The bottom left corner is the starting x-coordinate but with the ending y-coordinate and the bottom right corner is the ending x and y-coordinate. By adding and comparing all the small boxes coordinates, the extreme coordinates for the corners are acquired and used for the new box; and

3. At this stage, the bounding boxes are not yet drawn on the continuum image, and the process to merge the small boxes is repeated until there are no more small boxes close enough for merging into the new larger boxes. Then the final boxes are drawn from the new larger boxes. All the boxes are then drawn onto the continuum image and the resulting image is then saved.

The process of the Canny edge detection method is illustrated in Figure 1 and the result of the method is shown in Figure 2.

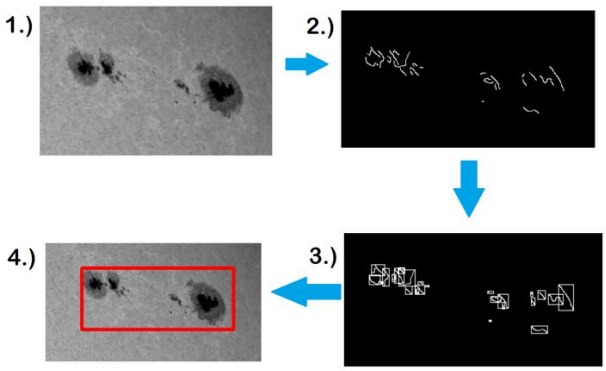

Figure 1. The process of the Canny algorithm

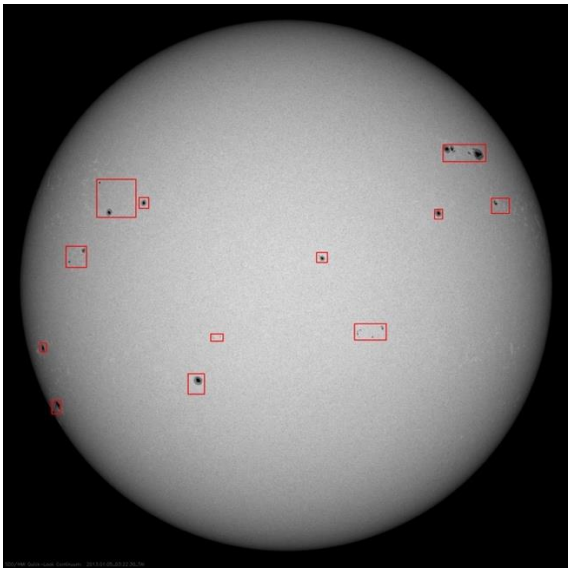

Figure 2. The result of the Canny method

2.2 Template Matching

An overview of the steps for identifying sunspots by using template matching is:

1. Identification of the sunspots by using the template matching algorithm of the OpenCV library to detect all the objects that resemble the templates of sunspots, on the images;

2. The same process to merge the boxes that were used for the Canny algorithm is also used here. The bounding boxes that will be merged were created from the resembling templates of sunspots that were detected; and

3. The bounding boxes are not yet drawn on the continuum image and step two is repeated until there are no more small boxes in merging distance, at which point the coordinates of the new larger boxes are also put through step two to get the final bounding boxes. All these boxes are then drawn on the image, which is subsequently stored to the file system.

A number of sunspots were extracted from images and used as templates for the template matching algorithm. These templates are shown in Figure 3.

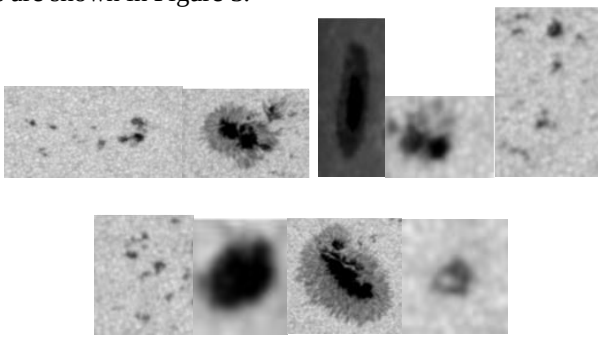

Figure 3. Templates used for template matching

2.3 Watershed

An overview of the steps for identifying sunspots by using watershed is:

1. The watershed algorithm uses the same implementation of the Canny algorithm for identifying sunspots;
2. The same merging that was done for both Canny and template matching is reused in the watershed algorithm;
3. Once the merging is done and the results of the Canny have been drawn on the image, the bounding boxes that were drawn are then extracted for watershed;
4. A default image of 2019-10-15 at 00:29 UT that contains no sunspots is subtracted from the extracted images. First, the same area of an extracted image is extracted from the default image and both are then blurred with a median blur. Both of the two extracted areas are subtracted and the result is saved. This is done for every extracted sunspot;
5. The subtracted image is then converted to binary with two different thresholds. The first threshold converts everything to white except that which is below the threshold; that is converted to black which is the umbra. The same principle is applied for identifying the penumbra where a threshold is applied to convert the penumbra area to black and the background to white; and
6. The subtracted penumbra and umbra images are then applied with the watershed algorithm. The penumbra is drawn with a red outline and the umbra with a green outline. The results of both images are then written on the original extracted image as well as on the original image. An extraction showing the process on a single sunspot and is shown in Figure 4.

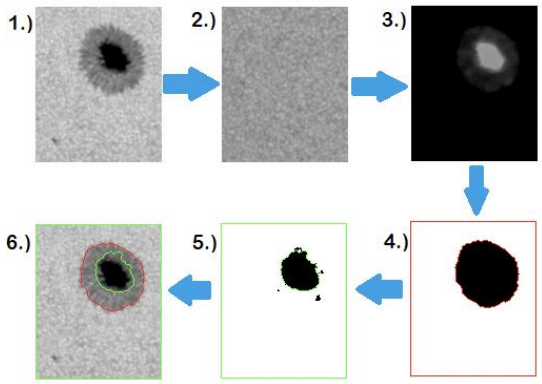

Figure 4. The result of watershed segmentation on a solar image

2.4 Algorithm for Tracking Sunspots

An overview of the steps for tracking sunspots is:

1. A starting image of the Sun is provided and sunspots are identified and bounding boxes are drawn using the steps discussed above;
2. The watershed algorithm is based on the results from the Canny algorithm that was implemented in the previous step. Thus, it uses the same implementation of the Canny algorithm for identifying the sunspots and merging the bounding boxes;
3. Assign an identification number to each of the bounding boxes;
4. Identify the sunspots on a second image, which will be used to compare to the starting image;
5. Calculate the time difference between images;
6. Determine the expected position of each of the sunspots identified on the starting image using the time elapsed multiplied by a predetermined coefficient. The coefficient is determined based on the range of the bounding box the starting x-coordinate is in.
7. Using this estimated position and a variance calculated using the time elapsed, find bounding boxes on the second image that are within range of estimated position +- variance.
8. The final step is to test if there are any bounding boxes close to the calculated distance from the starting image on the ending image. Depending on the time difference, the variance is also calculated to be larger for large time differences and the variance is less for smaller time differences between the images.

The process of tracking is illustrated in Figure 5 below. The starting image as well as the final image, 93 hours later, both show the sunspot group.

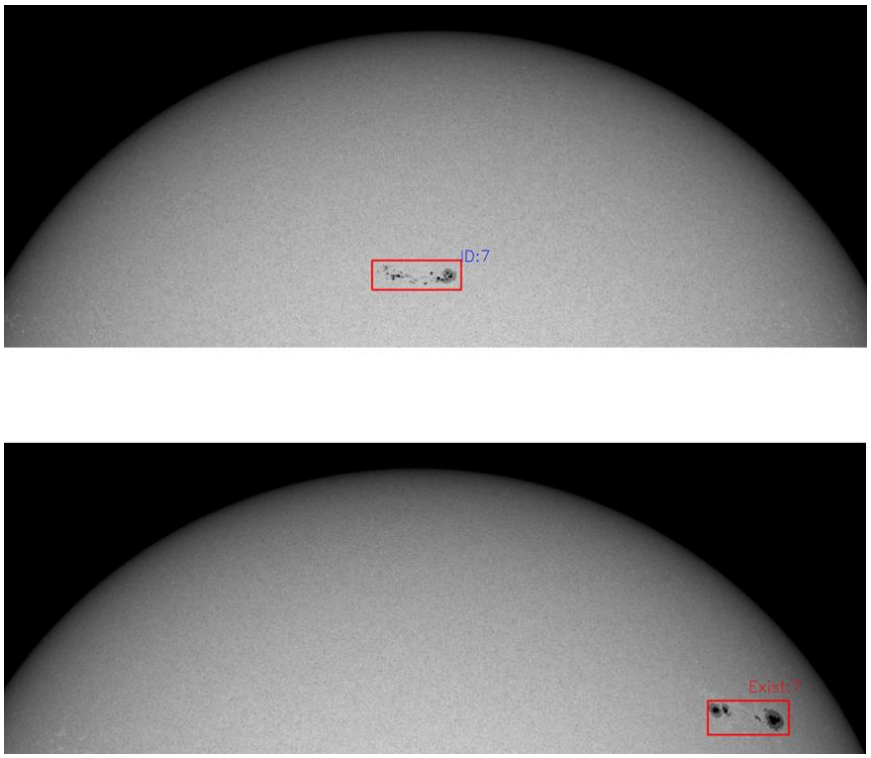

Figure 5. The result of the tracking algorithm. The top image that is the starting image for tracking with the date of 2013-01-01 starting at 06:00 UT. The bottom image is on 2013-01-05 at 03:00 UT in the morning. The bottom image is the result of tracking predicted at 93 hours later

3. RESULTS

The algorithms presented were tested on a number of solar images. These results of these tests will now be discussed.

Previous work done on sunspot detection was to produce a count of the number of sunspots on a given image. Furthermore, different time periods are used in different studies making it difficult to compare results with other studies. Also, the main aim of this study is the tracking of sunspots over a number of days. To be able to do this the sunspots first need to be identified and the success of these processes are therefore confirmed by visually comparing results with the images.

3.1 Canny Edge Detection

The results from the Canny algorithm proved to be accurate, in terms of detecting almost every sunspot as well as detecting a lot of pores on the Sun. A sunspot is counted when it has a clear penumbra as well as umbra area and a pore is classified as only small dark spots. When tested with an image that contains no sunspots, this algorithm does not detect any false positives on the image. From all the test images, 146 out of 153 sunspots were detected resulting in an error percentage of 5%. In only one case a false positive was detected and the sunspots that were missed were all on the edge of the Sun. The Canny method can be described as effective and accurate in detecting sunspots away from the limb.

3.2 Template Matching

The results from the template matching algorithm proved to be accurate, in terms of detecting almost every sunspot as well as detecting a lot of pores on the. A sunspot is counted when it has a clear penumbra as well as umbra area and pores are classified as small dark spots. Although, when tested with an image that contains no sunspots, this algorithm detects false positives that are not desired.

From all the test images, 144 out of 153 sunspots were detected resulting in an error percentage of 6%. In seven cases a false positive was detected and the sunspots that were missed were all on the edge of the Sun. The template matching method can be described as effective and accurate, but less accurate than the Canny method in detecting sunspots. The method was less sensitive in picking up small pores.

3.3 Watershed

For the purpose of being implemented on sunspots, the watershed algorithm was both sufficient and efficient. Additionally, it would be beneficial that more pores also be detected. This can be done by adjusting the threshold, but it was currently implemented in detecting the main focus of the images, which was the sunspots. The aim for this algorithm was satisfied with the results that were achieved.

3.4 Tracking

The tracking algorithm's results are given in Figures 5 and 6. It can be derived from the results below that the prediction of where the sunspots will be in the next image is highly accurate. It is important to note that this algorithm is based on the Canny algorithm's bounding boxes. Thus, if a sunspot is on the starting image but is not detected, and it is detected on the next image, it will be added as a new sunspot. Canny was chosen over template matching since it proved to more accurate and more stable in detecting true positives and not false positives such as template matching.

The only complex areas in this algorithm would be that if two sunspots are included in a single bounding box and in the next image the bounding box is split into two separate bounding boxes with the new box having a higher y-axis than predicted for. A close-up result of the algorithm's result is shown in Figure 2. An example of a single bounding box on the original image that is split into two bounding boxes on the next image, shows how the algorithm identifies both boxes with the same identification in Figure 6.

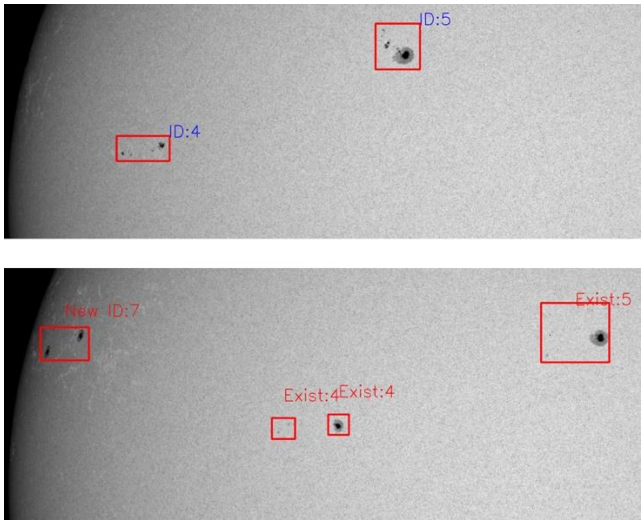

Figure 6. The result of the tracking algorithm where a bounding box is split into two boxes. The top image that is the starting image for tracking with the date of 2013-01-01 starting at 06:00 UT. The bottom image is on the 2013-01-02 at 21:00 UT in the evening. The bottom image is the result of tracking predicted at 39 hours later

4. CONCLUSIONS AND FUTURE WORK

The Canny algorithm proved to be the most accurate between it and template matching. Although, template matching drew the bounding boxes on separate sunspots where the Canny sometimes included both sunspots. The Canny algorithm also detected more pores than template matching making it much more sensitive to detecting not only the desired sunspots. When compared to using an image that contains no sunspots, the Canny method does not detect any false positives, whereas template matching detects false positives. The watershed algorithm performed efficient on large sunspots areas and the distinguishing between the umbra and penumbra was also efficient on the larger sunspots. Although, for better results between distinguishing the umbra and penumbra, pores and very small decaying sunspots that only has an umbra were not detected.

Tracking between a starting image and any following image also proved to be very accurate in predicting the position of the sunspots.

Future work that can be derived from this study includes:

1. The comparison against the accuracy for machine learning techniques in terms of identifying sunspots.
2. Investigation into algorithms that could help the prediction of CMEs could be incorporated or based on the results of this study.

REFERENCES

Colak, T. and Qahwaji, R., (2007). Automatic Sunspot Classification for Real-Time Forecasting of Solar Activities. *3rd International Conference on Recent Advances in Space Technologies*, Istanbul, Turkey, pp. 733– 738.

Curto, J. et al, (2008). Automatic Sunspots Detection on Full-Disk Solar Images using Mathematical Morphology. *Solar Physics*, Vol. 250, No. 2, pp 411–429.

Du Toit, R. (2022). Sunspot detection and tracking using image processing techniques. Master's thesis, North-West University, South Africa.

Fletcher, L. et al, (2011). An Observational Overview of Solar Flares. *Space Science Reviews*, Vol. 159, No. 1-4, pp 19–106.

Goel, S. and Mathew, S. K., (2014). Automated Detection, Characterization, and Tracking of Sunspots from SoHO/MDI Continuum Images. *Solar Physics*, Vol. 289, No. 4, pp 1413–1431.

Guarcello, M. G. et al, (2019). Simultaneous Kepler/K2 and XMM-Newton Observations of Superflares in the Pleiades. *Astronomy & Astrophysics*, Vol 622:A210.

Koskinen, H. et al, (2001). Space Weather Effects Catalogue. *ESA Space Weather Study (ESWS)*, Issue 2.2.

Manish, Tet al, (2014). Automatic Detection of Sunspot Activities using Advanced Detection Model. *IOSRJournal of Computer Engineerin*g, Vol. 16, No. 2, pp. 2278–8727.

Yu, L et al, (2014). Automated Sunspot Detection using Morphological Reconstruction and Adaptive Region Growing Techniques. *Proceedings of the 33rd Chinese Control Conference*, Nanjing, China, pp. 7168–7172.