Periodicals of Engineering and Natural Sciences ISSN: 2303-4521
Vol. 6, No. 1, April 2018, pp. 84 – 94
Available online at: *http://pen.ius.edu.ba*

Classification and Detection of Various Geographical Features from Satellite Imagery

Md. Haidar Sharif[1], Sahin Uyaver[2], and Zaid Zerdo[3]
[1]International Balkan University, Republic of Macedonia
[2]Turkish-German University, Turkey
[3]International University of Sarajevo, Bosnia and Herzegovina

Article Info

Article history:
Received August 2, 2017
Revised March 16, 2018
Accepted April 23, 2018

Keyword:
Classification
Craters
Detection
Satellite imagery

ABSTRACT

It is a challenging task to classify and detect various geographical features from the satellite imagery of the Earth as well as the celestial bodies. This paper puts forward several pixel based classification algorithms to classify geographical features from the satellite images of the Earth. The recorded experimental results, from a total of 606 satellite images to classify miscellaneous geographical features, demonstrate that the maximum algorithmic performances can approximate to 87%. This paper also addresses a simple algorithm based on edge approximation and circular Hough transformation to detect craters from the satellite imagery of celestial bodies. An online available dataset to detect craters evaluates the performance of the algorithm. In general, all the proposed algorithms are straightforward but in many ways effective.

Corresponding Author:
Name Md. Haidar Sharif
Affiliation International Balkan University
Address Skopje, Republic of Macedonia
Phone +38923214831
Email msharif@ibu.edu.mk

1. Introduction

It is a difficult work to classify and detect miscellaneous geographical features from the satellite imagery of the Earth as well as the celestial bodies. For example, Figure 1 shows sample satellite images of the Bosnian city of Banja-Luka [1]. We need to classify geographical features (e.g., cemeteries, fields, houses, industries, rivers, streets, forests, mountains, and so on) from those images. Cemeteries would contain lots of small and white patches or rectangles, while the industrial class might be quite varied and much more difficult to detect. Houses in the housing class may have red roofs, which would make the classification process a bit easy after extracting necessary features properly. But forests and rivers are not easy to classify because of their similar nature. Many researchers tried to solve the existing problem of classifying geographical features from the satellite images of the Earth with great accuracy [2–5]. For examples, Haralick et al. [2] addressed how texture plays a vital role in image classification of satellite imagery, photographic imagery, and photomicrograph of 1:20000 scales with the maximum accuracy of 83%. Cleve et al. [3] compared pixel and made object based classifications using high resolution aerial photographic images with an accuracy score of approximately 40%. Risojevic et al. [4] analyzed the importance of texture in remote image classification by employing the Gabor wavelet coefficients [6] and support vector machines [7] with an accuracy score of 85%. Risojevic [5] suggested how the images can be classified by dint of convolutional neural networks with the maximum accuracy of 99%.
Weih et al. [8] claimed that the object based classification is better than the pixel based classification. But in this paper, we are interested in pixel based classification algorithms - by motivating the proposition that the pixel based methods analyze the spectral properties of every pixel within the area of interest [9]. With this vein, we propose several pixel based classification algorithms to classify geographical features from the satellite images of the Earth. As experimental dataset, we utilize the satellite imagery of the Bosnian city of Banja-Luka to

Figure 1. Sample satellite images from Banja-Luka dataset [1]. How to classify the geographical features (e.g., cemetery, fields, houses, industry, river, and trees) from those images?

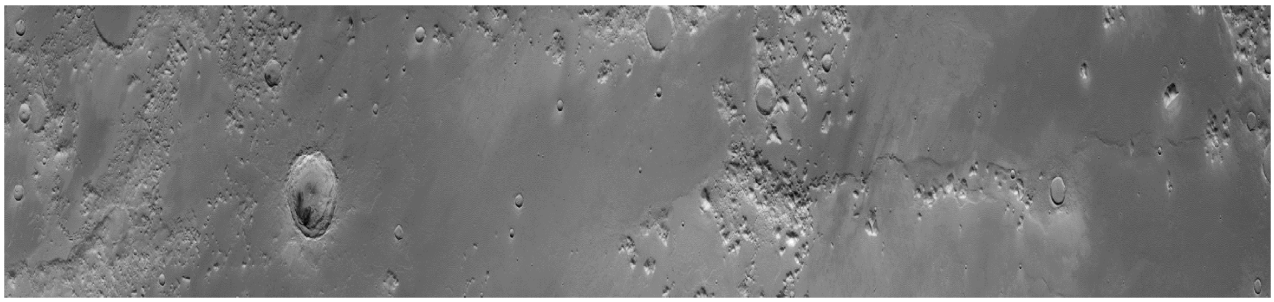


Figure 2. A sample satellite image of the Mars. How to detect the number of craters from this image?

classify geographical features. The Banja-Luka dataset [1] consists of 28 cemetery images, 178 field images, 143 houses images, 75 industry images, 77 river images, and 105 tree images. We also apply WEKA[1] to train neural networks for classifying geographical features.

A crater can be defined as a cavity on the surface of the Earth or any celestial body marking the orifice of a volcano. To provide ground truth of the craters in satellite imagery is a challenging task. If a dataset of satellite images from a celestial body is available, then one would interest to count the number of craters that exist in every image. For example, Figure 3 shows a sample satellite image of Mars. We wish to detect the number of existing craters on it. It is very hard to verify the algorithmic results if no ground truths are available. Besides, craters come in all shapes and sizes. Thus it is really a difficult task to detect them. The size is one of main problems, since some craters can be a few pixels wide while others can be hundreds of pixels wide. Craters can be inside of other craters. They can be cut in half due to the size of the image. Color of the craters can play a significant role. Distinct celestial bodies usually have different surface colors. For instance, the Mars is more orange and red in color, while its moon the Phobos [10] is grayer in color. Shades of craters play an important role, i.e., craters can contain shades due to neighboring hills or mountains. The detection of such craters is extremely hard. Another problem is that there exist no ground truths for the dataset of celestial bodies. Explicitly, no dataset ever mentions the exact number of craters. An approximate ground truth has been put for each dataset by using manual detection and counting. Like events detection (e.g., [11–13]), the crater detection became key interests of many computer vision researchers [14, 15]. For instance, Honda et al. [14] added a variant of the genetic algorithm to form a candidate of craters before using a self-organizing mapping to categorize the craters from non-craters. Meng et al. [15] constructed candidate areas that might contain craters using Hough transforms. The performance of their algorithm was not bad, but viability became a problem (e.g., false positive) when the craters did not contain a clear edge. Mu et al. [16] used cell algorithms to detect craters on celestial bodies with 86% accuracy. Upadhyay et al. [17] tried to detect the boundary of crater using functions from MATLAB.

We propose a simple algorithm based on edge approximation and circular Hough transform to detect craters from the satellite imagery of celestial bodies. It minimizes color and shape related problems of the craters. It is tested against an online available dataset [18] and its performance cannot be eschewed.

[1]Waikato Environment for Knowledge Analysis (WEKA) is a suite of machine learning software written in Java, developed at the University of Waikato, http://www.cs.waikato.ac.nz/ ml/. It provides a series of tools for pre-processing, classification, visualization, etc.

The rest of the paper is organized as follows. Section 2. illustrates necessary implementation steps. Section 3. reports experimental results followed by some hints for future work. Finally, Section 4. concludes the paper.

2. Implementation Steps

In this section, five key algorithms to classify satellite imagery and an algorithm to detect craters are discussed.

2.1. Algorithms to classify satellite imagery

2.1.1. P_{RGB} : Sum of Red-Green-Blue (RGB) pixels

The sum of all RGB pixel values for each image can be a straightforward algorithm to classify various geographical features from satellite images. The RGB pixel values of an image can distinguish between certain features quite easily. For example, cemetery images usually have large amounts of white pixels due to the gravestones, which could be seen from a high count of all RGB features. The river images do not contain many red pixels. But housing images possess a lot of red pixels because of their many red roofs. The images of field and forest have many green pixels, but they have much less red and blue pixels. Taking into account these common cases, a simple but somewhat effective classifier can be proposed.

2.1.2. P_{WGR} : White-Green-Red (WGR) pixels count

The addition of all RGB pixel values is not sufficient for a good classifier. Consequently, it is a good idea to count exactly how many WGR pixels exist in a given image. A white pixel cannot be considered as apparently red even though it contains a high level of red. Henceforth, it can be counted how many WGR pixels exist according to the following threshold conditions:

1. A pixel is considered as white if it contains RGB pixels greater than 200. And then the average of the three RGB pixel values is deemed.
2. A pixel is considered as green if it contains: a red pixel value smaller than 100, a blue pixel value smaller than 150, and a green pixel value greater than 100.
3. A pixel is considered as red if it contains: a red pixel value greater than 190, a green pixel value smaller than 200, and a blue pixel value not greater than 200.

The aforementioned constant values (i.e., 200, 100, 150, 100, 190, and 200) are experimentally defined to get good performance of the algorithm, so no theoretical reasoning is explained herewith.

2.1.3. D_T : Texture (T) difference in certain colors and images

Normally, there is no formal mathematical definition of texture. One of the first quantitative or physiological texture description can be found in [19], where six basic textural features, namely, coarseness, contrast, directionality, line-likeness, regularity, and roughness were approximated in computational. Bharati et al. [20] discussed various methods for image texture analysis. However, an image texture can be a series of metrics computed in image computation intended to measure the sensed texture of an image. Image texture provides knowledge about the spatial organisation of color or image intensities or exclusive domain of an image. The texture differences in certain colors and images can be implemented.

2.1.4. C_{WR} : White-Rectangle (WR) count

Due to the count of how many white shapes exist on a given image, the cemeteries are much easier to classify by employing WR count. A flood fill algorithm is implemented to go from one white pixel to another and then count the number of white pixels. The following steps for C_{WR} are used:

1. Convert an image to a gray image using a simple averaging method.
2. Convert the gray image into a binary image considering a simple thresholding technique that distinguishes pixels between extremely white, e.g., higher than 230 in its gray value, and non-white pixels.

3. Flood every fifth pixel with respect to width and height. The flooding is recursive with the stopping criteria being the edges of image, not going into another indexed flood fill and if it hits a non-white pixel.

4. A map is made that displays how many pixels should contain each index, i.e., how many pixels should each flood-fill make.

5. Indexes that have a moderate pixel count (e.g., between 5 and 30) are regarded as white rectangles.

The aforementioned constant values (i.e., 230, 5, and 30) are experimentally defined to get good performance of the algorithm, so no theoretical reasoning is explained hereby.

2.1.5. C_{RP} : River-Pixel (RP) count

One of the simplest ways to classify river images is to look at the individual pixels present in the image. The existing pixels on images of rivers are unique. The final status of a pixel whether it will be considered as a river pixel or not, it depends on the following a few sessions of analysis:

1. The absolute difference between green and blue values is negligible.

2. The absolute difference between green and red is considered.

3. The green value is considered.

2.1.6. Training of neural networks

WEKA is used to train neural networks. Its following classification methods to train neural networks are used.

1. Multilayer Perceptron (MulPer): It gives evidence of a neural network with feed forward capabilities and back propagation those results in a better learning outcome.

2. Logistic Regression (LogReg): It shows one of the possible regression models in which dependent variables are grouped but not continuous.

3. Sequential Minimal Optimization (SeqOpt): It makes clear and visible that one of the solutions to quadratic programming problems which presents in the training of support vector machines.

4. Random Forest (RanFor): It gives evidence of an algorithm in which a series of random decision trees are constructed and then giving statistical information such as mode or mean.

2.2. Algorithm to detect craters in celestial bodies

We have proposed a crater detection algorithm based on circular Hough transformation to detect craters from images of celestial bodies. The algorithm is simple but in many ways effective. The Hough transform is a technique that can be used to distinguish the parameters of a curve which best fits a set of given edge points. Under normal conditions, this edge description can be come into the possession of either a Roberts cross operator [21] or a Sobel filter [22] or a Canny edge detector [23]. Nevertheless, these edge detectors would be noisy and lacking in restraint as there exist multiple edge fragments corresponding to a single whole feature. The output of an edge detector exclusively ascertains the whereabouts of features in an image, and then Hough transform determines both kind of features and how many of them exist in the image. Notwithstanding, these properties of Hough transform give a rational assortment to recognize crater without circumscribing the possibilities to one astronomical object. It models candidate objects that resemble the looked for objects, whether they are perfectly or imperfectly shaped. The candidates are given preference to the basis as a local maximum in the parameter space. Both the standard Hough transform and the circular Hough transform are used in many computer vision applications. The standard Hough transform utilises the parametric representation of a line to detect straight line. Under normal conditions, its input is a binary image containing the edge pixels. During the transformation, it iterates through the edge points and calculate all e.g., angle, distance pairs for them. Each sinusoid curve belongs to a point. The crossing of two sinusoids suggests that their corresponding points belong to the same line. The

more sinusoids cross a given point, the more edge points are on the identical line. In spite of that, it can be taken into account the circular Hough transform to detect astronomical objects such as craters. The circular Hough transform relies on equations for circles. Consequently, it has three parameters namely the radius of the circle as well as the x and y coordinates of the center. A larger computation time and memory for storage are required to compute these parameters. As a result, it increases the complexity of extracting information from an image. For this reason, the radius of the circle would be fixed at a constant value. For each edge point, a circle can be drawn with that point as origin and radius. The circular Hough transform also uses a three-dimensional array, where the first two dimensions point out the coordinates of the circle and the last third specifying the radius. The values in the array are increased every time a circle is drawn with the desired radius over every edge point. The array, which kept counting of how many circles pass through coordinates of each edge point, proceeds to a vote to find the highest count. The coordinates of the center of the circles in the images are the coordinates with the highest count. The Listing 1 illustrates our proposed crater detection algorithm.

Listing 1. Our proposed crater detection algorithm

```
rgbI = imread('Image0001.png'); % Get a camera view RGB image.
figure
imshow(rgbI); % The I be a 3D image but we need 2D image.
[height,width,dimension] =size(rgbI);
x = sprintf('An original camera view image of size %d x %d x %d',height,width,dimension);
set(gca,'fontsize',21); % Set font size.
disp(title(x)); % Dsplay
grayI = rgb2gray(rgbI); % Convert from 3D to 2D using a weighted mean of RGB channels.
figure
imshow(grayI);
[height,width,dimension] =size(grayI);
x = sprintf('Converted grayscale intensity image of size %d x %d x %d',height,width,dimension);
set(gca,'fontsize',21); % Set font size.
disp(title(x));
bi = edge(grayI,'canny'); %Create a binary image of grayI using Canny approximation.
figure
imshow(bi);
set(gca,'fontsize',21); % Set font size.
title('The found edges in the grayscale intensity image using Canny approximation.');
radii = 10:1:30; % Search radii from 10 to 30 pixels in steps of 1 pixel.
[h, margin]=circle_hough(bi,radii,'same','normalise'); %Hough transform for circles.
figure
set(gca,'fontsize',21,'LineWidth',3,'Color','green');
for i=1:length(h(1,:))
    if ((h(1,i)~= 0) || (h(2,i)~= 0)|| (h(3,i)~= 0))
    x(i) =  h(1,i);
    y(i) =  h(2,i);
    r(i) =  h(3,i);
    d(i) =  r(i)*2;
    px(i) = x(i)-r(i);
    py(i) = y(i)-r(i);
    rectangle('Position',[px(i) py(i) d(i) d(i)],'Curvature',[1,1]);
    daspect([1,1,1]);
    end
hold on
xlabel('x-coordinate');
ylabel('y-coordinate');
title('The circles on Hough transform image.');
end
peaks = circle_houghpeaks(h,radii,'npeaks',13); % Select any number (say 13) of prominent peaks.
figure
set(gca,'fontsize',21);
xy = [peaks(1,:)', peaks(2,:)'];
hist3(xy,[max(peaks(3,:))  max(peaks(3,:))]);
set(get(gca,'child'),'FaceColor','interp','CDataMode','auto'); % Color the bars based on height.
xlabel('x-pixel values');
ylabel('y-pixel values');
zlabel('Radii values');
title('13 prominent peaks on the Hough transform image.');
peaks = circle_houghpeaks(h,radii,'npeaks',2); % Get any number (say 2) of most prominent peaks.
figure
set(gca,'fontsize',21);
```

```
imshow(rgbI);
hold on;
for peak = peaks
    [x,y] = circlepoints(peak(3));
    plot(x+peak(1), y+peak(2), 'r-','LineWidth',5,'Color','red');
end
title('The most prominent 2 peaks have been located on the original image.');
hold off
```

The *circle_hough(bi, radii,'same','normalise')* is one of the MATLAB (MATrix LABoratory) built-in functions. It takes a binary two dimensional image bi and a vector *radii* giving the radii of circles to detect. It returns the three dimensional accumulator array *h*, and an integer *margin* such that *h(i,j,k)* contains the number of votes for the circle centred at *bi(i-margin, j-margin)*, with radius *radii(k)*. Circles which pass through *bi* but whose centres are outside *bi* receive votes. Each feature in *bi* is allowed 1 vote for each circle. The option *same* returns only the part of h corresponding to centre positions within the image. In this case *h(:,:,k)* has the same dimensions as *bi*, and *margin = 0*. This option should not be used if circles whose centres are outside the image are to be detected. The option *normalise* multiplies each slice of *h*, *h(:,:,k)*, by *1/radii(k)*. This may be useful because larger circles get more votes, roughly in proportion to their radius. The *circle_houghpeaks(h,radii,'npeaks',15)* be another MATLAB built-in function. It locates the positions of 15 peaks in the return of *circle_hough* function. The result *peaks* is a $3 \times N$ array, where each column gives the position and radius of a possible circle in the original array. The first row of *peaks* has the x-coordinates, the second row has the y-coordinates, and the third row has the *radii*. Assume that Figure 3 (a) demonstrates an image of the Moon surface. We are interested to look for specific patterns which exist on its surface. Especially, there exists a visual crater in it. How can we detect that? The images of Figure 3 (b)-(f) depict the algorithmic output of Listing 1 to detect craters. Figure 3 (f) displays that our algorithm in Listing 1 detected correctly the existing two craters in Figure 3 (a).

3. Experimental Results

3.1. Experimental setup

Primarily, we have performed experiments on a computer of 8-Core CPUs at 3.50 GHz with 16 GB RAM. WEKA was used to train neural networks. Satellite images of both Banja-Luka dataset [1] and one of the online available crater datasets [18] were used to verify the performance of our algorithms. The number of hidden layers in the MulPer was considered as 3. The default values of WEKA algorithms are used for other parameters, e.g., learning rate and configuration of MLP (number of neurons per layer).

3.2. Results of geographical features classification

Table 1 depicts the results of our algorithms to classify various geographical features from Banja-Luka dataset without using the classification algorithms of WEKA. Algorithm C_{RP} performed the best average of 86.66% among its alternative individual algorithms of P_{RGB}, P_{WGR}, D_T, and C_{WR}. This is widely due to the images of fields, houses, industries, rivers, and trees highly matched the following three cases: (i) The absolute difference between green and blue values has been neglected; (ii) The absolute difference between green and red has been taken in account; (iii) The green value has been taken in account. But those conditions did not suit very well for the case of cemetery images. Besides, cemeteries have huge amounts of white pixels because of gravestones. The number of white pixels is very high as compared to the number of other RGB pixels. Thus the algorithm C_{RP} resulted the worst performance (78%) in the case of cemetery classification. The combined algorithm P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} demonstrated the best performance as 86.83% from Banja-Luka dataset without using the classification algorithms of WEKA. Notwithstanding, from Table 1 it is noticeable that the average performance of the algorithm P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} without using the classification algorithms of WEKA is slightly greater than that of C_{RP}.

Table 2 presents the performances of our various algorithms to classify various geographical features from Banja-Luka dataset using the classification algorithms of WEKA. The gain with respect to the P_{RGB} algorithmic performance in Table 2 indicates the performance increment or decrement when we add or remove any new feature to the existing features of an algorithm. Approximately 70% average success rate was resulted in the cases of individual algorithms either P_{RGB} or P_{WGR} or D_T or C_{WR} or C_{RP}. But the combined algorithm

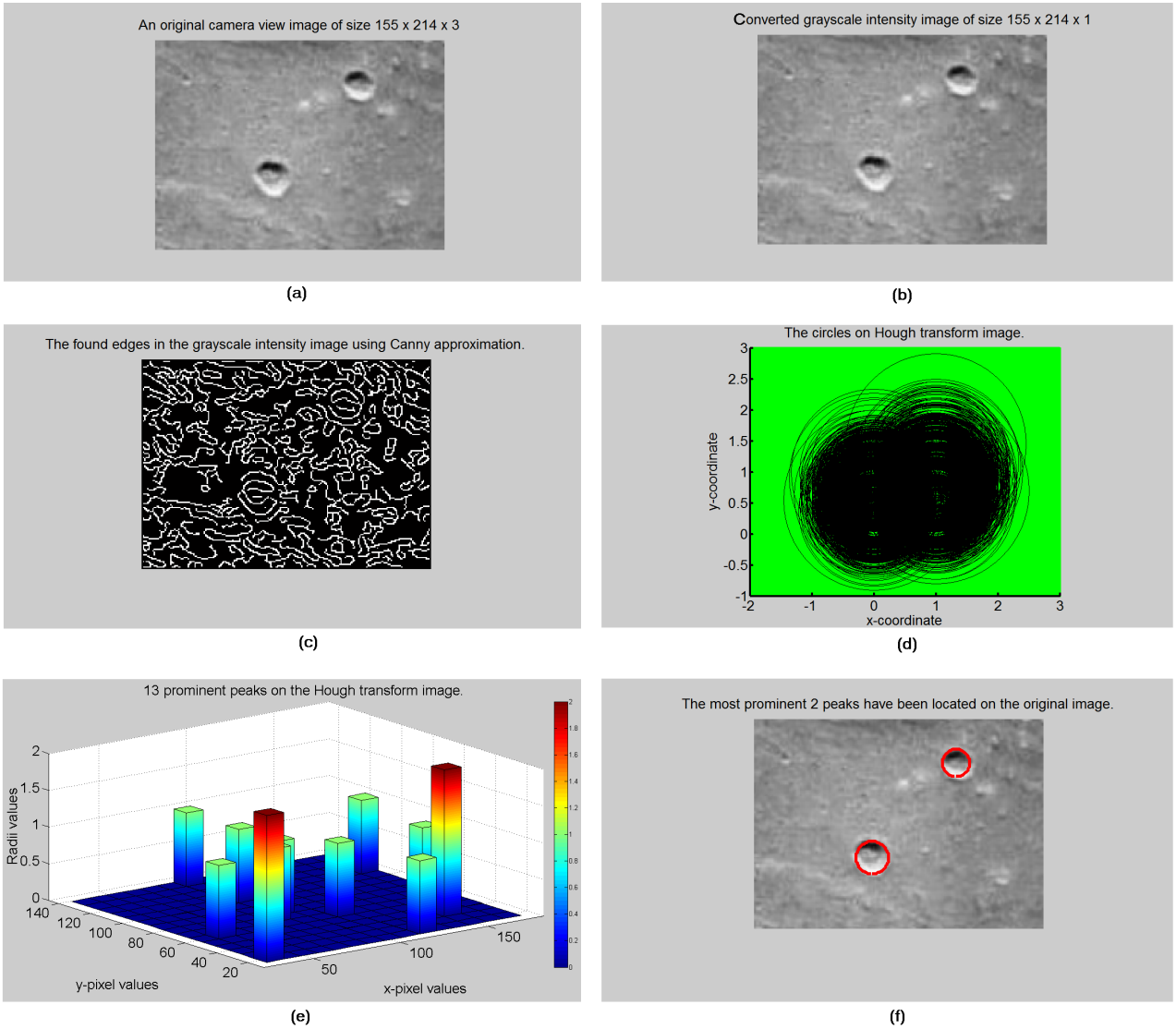


Figure 3. A sample output of the algorithm on the Listing 1.

P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} performed the best success rate of 87.34% from Banja-Luka dataset using the classification algorithms of WEKA. Using the sequential minimal optimization of WEKA classification algorithm, the gains of the algorithms of P_{WGR}, D_T, and C_{WR} were decreased with the amount of 1.45%, 3.37%, and 2.28%, respectively. But the gain of C_{RP} was increased by 13.16%. The gains of the algorithms P_{WGR}, D_T, C_{WR}, and C_{RP} were not reported for multilayer perceptron, logistic regression, and random forest. In case of the algorithm of P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP}, the gains were increased up to 30.18%, 21.44%, 21.48%, and 26.67% for sequential minimal optimization, multilayer perceptron, logistic regression, and random forest, respectively. The comparative performance factor for the algorithm of P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} without using WEKA or using WEKA is not significant. This is widely due to the fact that P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} without using WEKA is already presented its optimal performance. Consequently, no optimization options have significant effect on its further performance increment. Normally, performance would be increased if WEKA classification algorithms are used. But this proposition may not be true for optimal cases. For example, the performance would be decreased if WEKA classification algorithms are used. The average performance of the algorithm C_{RP} without using WEKA classification algorithms was 86.66% but after using WEKA classification algorithms its performance became to 70%, i.e., approximately 19.22% performance degradation. Similarly, about 8% mean performance degradation can be observed in case of the algorithm P_{RGB}.

As a concluding remark we would say, the overall performance of our proposed algorithms demonstrate that

Table 1. Performance of our algorithms to classify various geographical features without using WEKA.

Algorithms	Items and their qualitative analysis of performance						
	Cemeteries	Fields	Houses	Industries	Rivers	Trees	Average
P_{RGB}	69%	72%	77%	80%	73%	86%	76.17%
P_{WGR}	65%	68%	71%	60%	62%	67%	65.50%
D_T	58%	53%	55%	47%	58%	64%	55.83%
C_{WR}	70%	64%	65%	64%	69%	60%	65.33%
C_{RP}	78%	94%	95%	84%	79%	90%	86.66%
P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP}	84%	91%	92%	83%	81%	90%	86.83%

Table 2. Performance of our algorithms to classify various geographical features using WEKA.

Algorithms	Normalized performance of numerous classification algorithms								
	SeqOpt		MulPer		LogReg		RanFor		Mean
	Success	Gain	Success	Gain	Success	Gain	Success	Gain	Value
P_{RGB}	62.62%	0	73.78%	0	72.81%	0	70.87%	0	70.02%
P_{WGR}	61.71%	-1.45%	73.78%	0	72.81%	0	70.87%	0	70.02%
D_T	60.51%	-3.37%	73.78%	0	72.81%	0	70.87%	0	70.02%
C_{WR}	61.19%	-2.28%	73.78%	0	72.81%	0	70.87%	0	70.02%
C_{RP}	70.86%	13.16%	73.78%	0	72.81%	0	70.87%	0	70.02%
P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP}	81.52%	30.18%	89.60%	21.44%	88.45%	21.48%	89.77%	26.67%	87.34%

they can play some sort of important role in the machine learning and computer vision applications including classification of miscellaneous geographical features from satellite imagery.

3.3. Comparison with state-of-the-art method

The method of Risojevic et al. [4] produced 85% correct classification results from Banja-Luka dataset. The average performances of our proposed algorithms C_{RP} and P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} as well as the algorithmic performance of Risojevic et al. [4] have been demonstrated in Table 3. It appears to exist that our simple algorithms having up to 87% correct classification results function more or less in parallel with the state-of-the-art method (e.g., Risojevic et al. [4]) without any serious concern of effectiveness.

Table 3. Performance comparison for classifying various geographical features from Banja-Luka dataset.

Algorithms	Success Rate
Risojevic et al. [4]	85.00%
C_{RP} without WEKA [Ours]	86.66%
P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} without WEKA [Ours]	86.83%
P_{RGB}+P_{WGR}+D_T+C_{WR}+C_{RP} using WEKA [Ours]	87.34%

3.4. Results of craters detection

Figure 4 depicts a sample of input image (a), edge detection (b), circular Hough transformation (c), and the best performance for the detection of craters (d) using the algorithm as depicted in the Listing 1. Figure 5 displays a sample of input image (a), edge detection (b), circular Hough transformation (c), and the worst performance for the detection of craters (d) using the algorithm as depicted in the Listing 1. It seems that our crater detection algorithm worked well in many case but with modest average results. In general, many craters were detected

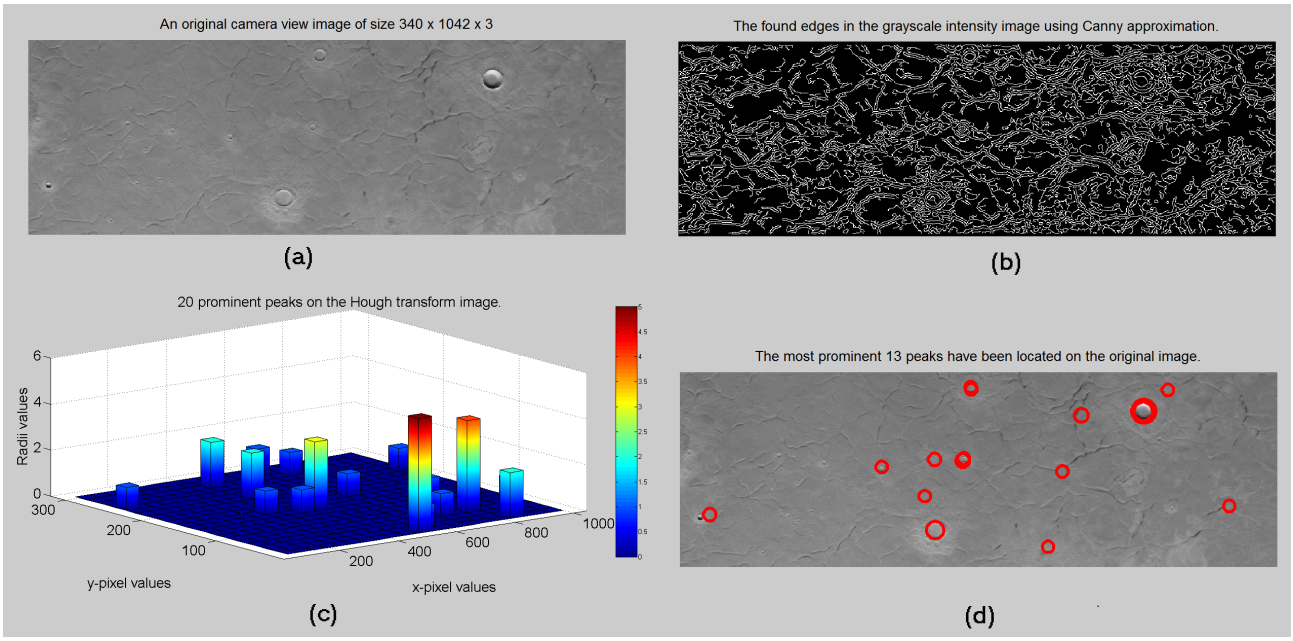


Figure 4. A sample of one of the best detection results of craters using our proposed algorithm.

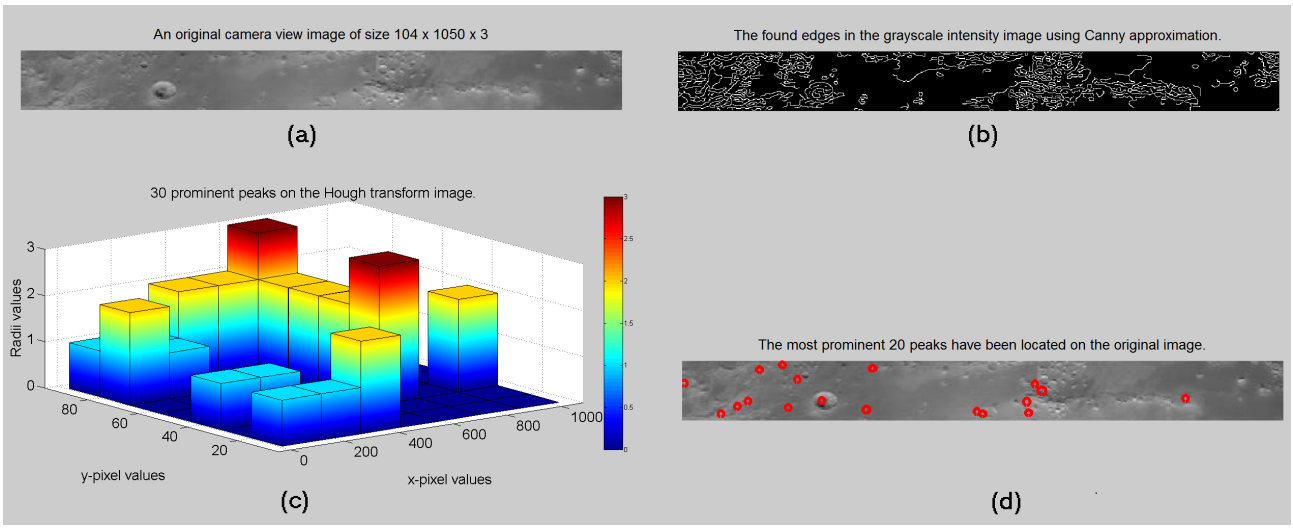


Figure 5. A sample of one of the worst detection results of craters using our proposed algorithm.

correctly by our proposed algorithm. But small craters, the craters which were on the border, as well as many non-crater-like objects made a serious challenge of our algorithm.

3.5. Future works

Our straightforward classification algorithms need less processing time to extract simple features. On the average, their accuracy was recorded more than 70% from the Banja-Luka dataset. This performance would arouse attention for many applications related to computer vision and pattern recognition. But more advanced features (e.g., features regarding texture and object recognition) could be extracted. If more features would be extracted and utilized, then dimensionality reduction techniques (e.g., principal component analysis) should be employed to find exactly which features will not affect the solution. Only 606 satellite images were used to test our classification algorithms. It is important to test them with a large series of other datasets. More data means more possible observations to make in order to enhance the algorithm. To improve the algorithmic results further, the algorithms would need to be analyzed and their functionalities learnt to see which model fits datasets the best. Areas under the receiver operating characteristic curves [13] and multiple comparisons with statistical

tests [24] would be performed by considering the proposed algorithms and state-of-the-art algorithms. Time complexity [25] of each algorithm with cache memory effect [26, 27] would be computed and compared.

4. Conclusion

Our first aim was to classify geographical features of a large dataset with high-resolution images in a short amount of time. With this vein, we proposed several pixel based classification algorithms to classify geographical features from the satellite images of the Earth. The recorded experimental results of the algorithms from the Banja-Luka dataset showed their highest effectiveness approximately 87%. Our second aim was to detect craters from the images of the other planets rather than the Earth by proposing a simple but effective algorithm. Its output from an online available crater dataset showed the partial fulfilment of our aim. Craters come in different colors, shapes, and sizes. Our algorithm minimizes the problems related to surface color and shape of the craters. But the size problem of craters remains for the future investigation. Future study would also include further improvement of the algorithmic performances along with error analysis and time complexity.

REFERENCES

[1] V. Risojevic, "Aerial image classification : Database of RGB+NIR aerial images," 2017, retrieved from http://dsp.etfbl.net/aerial.

[2] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, no. 6, pp. 610–621, Nov 1973.

[3] C. Cleve, M. Kelly, F. R. Kearns, and M. Moritz, "Classification of the wildland–urban interface: A comparison of pixel-and object-based classifications using high-resolution aerial photography," *Computers, Environment and Urban Systems*, vol. 32, no. 4, pp. 317–326, 2008.

[4] V. Risojevic and Z. Babic, "Orientation difference descriptor for aerial image classification," in *International Conference on Systems, Signals and Image Processing (IWSSIP)*, April 2012, pp. 150–153.

[5] V. Risojevic, "Analysis of learned features for remote sensing image classification," in *Symposium on Neural Networks and Applications (NEUREL)*, 2016, pp. 1–6.

[6] T. S. Lee, "Image representation using 2D Gabor wavelets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959–971, 1996.

[7] F. Galip, M. H. Sharif, M. Caputcu, and S. Uyaver, "Recognition of objects from laser scanned data points using svm," in *International Conference on Multimedia and Image Processing (ICMIP)*, 2016, pp. 28–35.

[8] R. C. Weih and N. D. Riggan, "Object-based classification vs. pixel-based classification: comparitive importance of multi-resolution imagery," in *Geographic Object-Based Image Analysis*, 2010, pp. 1–6.

[9] D. Enderle and R. Weih, "Integrating supervised and unsupervised classification methods to develop a more accurate land cover classification," *Journal of the Arkansas Academy of Science*, vol. 59, pp. 65–73, 2005.

[10] NASA, "Mar's moon Phobos," July 16, 2016, retrieved from https://mars.nasa.gov/allaboutmars/extreme/moons/phobos/.

[11] M. H. Sharif, N. Ihaddadene, and C. Djeraba, "Finding and indexing of eccentric events in video emanates," *Journal of Multimedia*, vol. 5, no. 1, pp. 22–35, 2010.

[12] M. H. Sharif and C. Djeraba, "An entropy approach for abnormal activities detection in video streams," *Pattern Recognition*, vol. 45, no. 7, pp. 2543–2561, 2012.

[13] M. H. Sharif, "An eigenvalue approach to detect flows and events in crowd videos," *Journal of Circuits, Systems, and Computers*, vol. 26, no. 7, pp. 1–50, 2017.

[14] R. Honda, O. Konishi, R. Azuma, H. Yokogawa, S. Yamanaka, and Y. Iijima, "Data mining system for planetary images-crater detection and categorization," in *Proceedings of the International Workshop on Machine Learning of Spatial Knowledge in conjunction with ICML, Stanford, CA*, 2000, pp. 103–108.

[15] D. Meng, C. Yunfeng, and W. Qingxian, "Method of passive image based crater autonomous detection," *Chinese Journal of Aeronautics*, vol. 22, no. 3, pp. 301–306, 2009.

[16] Y. Mu, W. Ding, D. Tao, and T. F. Stepinski, "Biologically inspired model for crater detection," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 2487–2494.

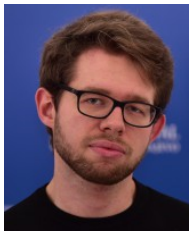
[17] A. Upadhyay and P. Ray, "Crater boundary detection and size frequency distribution (SFD) using MATLAB," Indian Space Research Organization, Bangalore, India, Project Report 116, 2011.

[18] HiRISE, "High resolution imaging science experiment, HiRISE Operations Center, Tucson, Arizona, United States of America," 2017, retrieved from https://www.uahirise.org/catalog.
[19] H. Tamura, S. Mori, and T. Yamawaki, "Textural features corresponding to visual perception," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 8, no. 6, pp. 460–473, 1978.
[20] M. H. Bharati, J. Liu, and J. F. MacGregor, "Image texture analysis: methods and comparisons," *Chemometrics and Intelligent Laboratory Systems*, vol. 72, no. 1, pp. 57 – 71, 2004.
[21] L. G. Roberts, "Machine perception of three-dimensional solids," Ph.D. dissertation, Dept. of Electrical Engineering, Massachusetts Institute of Technology, 1963, retrieved from http://www.packet.cc/files/mach-per-3D-solids.html.
[22] I. Sobel, "On calibrating computer controlled cameras for perceiving 3-D scenes," *Artificial Intelligence*, vol. 5, no. 2, pp. 185–198, 1974.
[23] J. F. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.
[24] H. Kusetogullari, M. H. Sharif, M. S. Leeson, and T. Celik, "A reduced uncertainty-based hybrid evolutionary algorithm for solving dynamic shortest-path routing problem," *Journal of Circuits, Systems, and Computers*, vol. 24, no. 5, 2015.
[25] M. H. Sharif, "A numerical approach for tracking unknown number of individual targets in videos," *Digital Signal Processing*, vol. 57, pp. 106–127, 2016.
[26] K. Khanfar, "An effective LRU with random replacement policy for cache memory," in *International Conference on Parallel and Distributed Processing Techniques and Applications*, 2003, pp. 1837–1843.
[27] M. H. Sharif, "High-performance mathematical functions for single-core architectures," *Journal of Circuits, Systems, and Computers*, vol. 23, no. 4, 2014.

BIOGRAPHY OF AUTHORS

Md. Haidar Sharif obtained his BSc, MSc, and PhD from Jahangirnagar University (Bangladesh), Universität Duisburg-Essen (Germany), and Université Lille 1 (France) in the years of 2001, 2006, and 2010, respectively. He had been working as an Assistant Prof at Bakırçay Üniversitesi in Turkey from January 2011 to January 2016. He had been working as an Assistant Prof at International University of Sarajevo in Bosnia and Herzegovina from April 2016 to June 2017. He has been working as an Associate Prof at International Balkan University in the Republic of Macedonia since September 2017. His research interests include computer vision and computer architecture.

Sahin Uyaver obtained his BSc and MSc in Physics from Yıldız Teknik Üniversitesi in the years of 1992 and 1996, respectively. He got his PhD from Universität Potsdam (Max Plack Institute of Colloids and Interfaces) in 2004. He had been working as an Assistant Prof at Maltepe Üniversitesi from 2005 to 2009, İstanbul Ticaret Üniversitesi from 2010 to 2014, and University of Oklahoma from 2014 to 2015. He has been working as an Associate Prof at Türk-Alman Üniversitesi since 2015. His research interests include polymers in solution and at interfaces, brain-computer interface, applications of computer vision and pattern recognition, and artificial intelligence.



Zaid Zerdo obtained his Bachelor of Science (BSc) in Computer Sciences and Engineering from International University of Sarajevo (IUS) in 2016. From September 2016 to June 2017, he had been working as a teaching assistant at IUS. He has been working as an Engineer at AtlantBH, one of the leading software development and outsourcing companies, in Bosnia and Herzegovina since July 2017. He has been studying his MSc at IUS since September 2017 on the part time basis. His research interests include web application and development, computer vision and image understanding, artificial intelligence, geographic information system and mapping, big data, and astronomy.