

Edge detection in Cassini astronomy image using Extreme Learning Machine

Xiqi Yang¹, Qingfeng Zhang^{1,2,*}, Xiyu Yang³, Qingyu Peng^{1,2}, Zhan Li^{1,2}, and Na Wang^{1,2}

¹Department of Computer Sciences, Jinan University, Guangzhou, 510632, P. R. China

²Sino-French Joint Laboratory for Astrometry, Dynamics and Space Science, Jinan University, Guangzhou, 510632, P. R. China

³School of Electrical and Information, Jinan University, Zhuhai, 519070, P. R. China

Abstract. Edge detection is often performed on disc-like object in Cassini astronomy images to accurately obtain the object's center position. The existing edge extraction methods usually produce lots of false edge pixels because of noise and the interior details in disc-like objects. In the paper, an edge detection algorithm based on Extreme Learning Machine (ELM) is proposed for Cassini astronomy images. In the ELM model, a 28-D feature vector of a pixel in Cassini image is constructed as input, which consists of first and second derivatives and some Haar-like features, and a binary classifier is obtained as output that tells if the pixel is in edge. The experimental result shows that its performance is much better than traditional operators. The detected edge is closer to the actual contour. Its average accuracy is 0.9379. The algorithm can be applied to edge detection of disc-like objects in astronomy images.

1 Introduction

Edge detection plays an important role in the processing of astronomy images. In the analysis of astronomy images, researchers often perform edge detection on disc-like object (such as images of elliptical or irregularly shaped observation objects), and then analyze their physical properties. Basing on different purposes and different situations, researchers often use different methods for edge detection. Young CA et al. [1] used the multi-scale wavelet transform to detect the edge of the solar corona and analyze the physical properties of the sun. Liu A et al. [2] and Saheba SM et al. [3] used the multi-scale adaptive Canny algorithm to detect the moon crater. Cornet T et al. [4] used the gradient method to detect the edge of Ontario Lacus on the Titan taken by Cassini to examine the change of the lake's edge. Walker SA et al. [5] used a Gaussian gradient filter to highlight the cold front edge of the galaxy.

The Cassini space probe has taken about one million astronomy images, most of which require using edge detection to analyze. When edges are detected by using the traditional edge detection algorithm [6-8], many false detections are brought due to noise pollution, which makes it difficult to further explore the physical properties of the observed objects. With the development of machine learning technology, many studies have introduced

* Corresponding author: tqfz@jnu.edu.cn

machine learning techniques for edge detection [9-12]. However, these techniques only use nature images or specific images like medical images and remote sensing images as the training set thus do not apply well to Cassini astronomy images. For Cassini astronomy images, detecting edge pixels accurately without redundancy is a problem to be solved. Therefore, this paper proposes an edge detection method using Extreme Learning Machine (ELM) for Cassini astronomy images.

The paper is divided into five parts. The second part introduces the edge detection algorithm based on the ELM model. The third part describes the establishment and training of the ELM model. The fourth part illustrates the experimental results and the fifth part makes a conclusion.

2 Methodology

The edge detection method based on the ELM model uses the trained ELM model to classify each pixel in the image (class 0 represents a non-edge while class 1 represents an edge). The entire process includes image preprocessing, feature extraction, ELM model building, model training, and using models to classify pixels. One of the keys of the process is the building and training of ELM model.

2.1 Image preprocessing

Since astronomy images are often interfered with by various noise signals in the process of acquisition and transmission, preprocessing is required before edge detection. This paper uses morphological transformation (erosion and dilation) to improve the edge connection and bilateral filtering to reduce noise and maintain edges.

2.2 Feature extraction

Feature extraction is an important basis for accurately determining whether a pixel is an edge pixel. After experiments, we finally extracted 28 features belonging to 3 types for each pixel.

2.2.1 First-order gradient (nine features)

First-order gradient is commonly used in edge detection. Here we have extracted first-order gradients $g(i)$ ($i=1, 2, \dots, 8$) in eight directions (namely $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$) in the eight neighborhood of each pixel. In addition, the gradient amplitude ga is used, which is calculated as follows:

$$ga = \sum_{i=1}^8 g(i) \quad (1)$$

2.2.2 Second-order gradient (nine features)

Second-order gradient is essentially the rate of change of the brightness gradient. In the ideal case of continuous change, detecting zero crossings in the second derivative will get a local maximum of the gradient. Different from the traditional calculation method of second-order gradient, we compute the second-order derivatives in eight directions in one pixel's 5×5 neighborhood. In addition, the second-order gradient amplitude is also calculated according to the formula (1).

2.2.3 Haar-like feature (ten features)

The Haar-like feature is a simple rectangular feature introduced in the face detection system by Viola et al. [13-14] and named after the Haar wavelet. Lienhart R et al. [15] extended it further by adding rectangular features with a rotation of 45°. The extended features are roughly divided into three types: edge features, line features, center-surround features (as shown in Fig. 1). The Haar-like feature can effectively reflect the local gray change information of the image, and can also be quickly calculated through the integral image.

In this paper, we select linear features and center-surround features as templates. The linear features Fig.1 (2a)-(2b) are computed in the dimensions of 2×3 and 2×4, respectively. They are rotated at 45° and 90° to get new features Fig.1 (2c)-(2h). The feature Fig.1 (3a) uses a 3×3 window, which is also rotated through 45° to get feature Fig.1 (3b). Finally, it is calculated using the integral image and 10 feature values are extracted for each pixel.

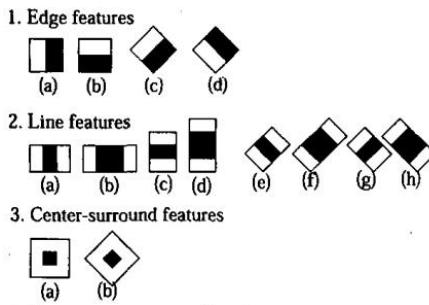


Fig. 1. Haar-like features. (1) Edge features. (2) Line features. (3) Center-surround features.

So far, we have extracted nine first-order gradient features, nine second-order gradient features, and ten Haar-like features to describe a pixel’s character. In total, a 28-dimensional feature vector is constructed for each pixel.

2.3 Extreme learning machine

Extreme Learning Machine (ELM) is a single-layer forward feedback neural network proposed by Huang et al. [16]. The input weights and hidden layer biases are all generated by program randomization instead of setting artificially. The output weights are determined by least squares regression. Therefore, the whole training process only needs to specify the number of hidden layer neurons and the activation function. Compared with the traditional neural network algorithm, the training speed is faster and the parameter selection is more flexible. At present, ELM has demonstrated its superior performance in many fields [17-19].

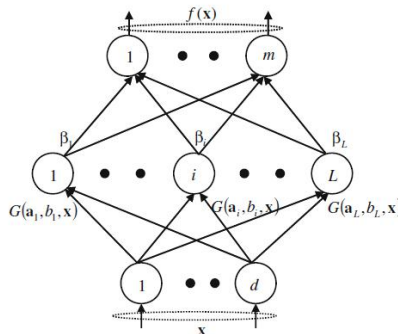


Fig.2. Single-layer ELM structure. ($m=2, L=20, d=28$).

The hierarchical structure of the ELM is shown in Fig. 2. From top to bottom, the output layer, hidden layer, and input layer are respectively shown. We use the 28-dimensional feature vector as input, and finally output the 2-D vector. The numbers of nodes in input layer, hidden layer and output layer are 28, 20 and 2, respectively. The activation function is the sigmoid function.

3 Training set preparation and model Training

To get the right weights and bias in ELM model, we need train it by data set, so preparing the training set is first phase for training model, which includes the extraction of the region of interest (ROI), sample labeling, and sample equalization.

3.1 Training set preparation

3.1.1 Extract region of interest (ROI)

Since the size of the astronomy image in our training set is 1024×1024 , which is much larger than the size of the celestial body in the image, if the entire image is directly input into the neural network, the training effect may be greatly reduced. Therefore, the first step of the experiment is to extract the region of interest (ROI).

3.1.2 Semi-automatic labeling

In our experiment, the training set and testing set are semi-automatic labeled using the manual annotation and the Canny operator. First, after the edges are detected with the Canny operator, the non-contour pixels are manually removed. Then, check whether there are edge points not marked by the canny operator and label the missed edge pixels.

3.1.3 Sample equalization

After ROI extraction and sample labeling, the resulting samples are divided into two classes: positive samples (edge pixel samples) and negative samples (non-edge pixel samples). Without further processing, the proportion of positive and negative samples is close to 1:400. The proportion of the original data sample in this paper is obviously out of balance.

For data imbalance problems, common solutions include under-sampling, over-sampling, introducing cost-sensitive factors into the algorithm [20-21] and so on. Here, we adopt the method of under-sampling as well as the weighted performance measures. Through reducing the size of the majority class by randomly removing some samples of the majority class, we finally make the ratio of positive and negative samples close to 1:4.

3.2 Model training

In the training phase, the equalized training data is input into a single-layer neural network based on Extreme Learning Machine (ELM). The ELM randomly initializes the input weights and hidden layer biases, and make input neurons and the hidden layer neurons fully connected. Then the output weights are calculated by least squares regression, and finally the hidden layer neurons are fully connected with the output neurons to output the classification results. If the classification accuracy rate and the classification error rate does not reach a certain threshold and the number of iterations does not reach 100, the above

training process is repeated. When the training is finished, a classifier for distinguishing edge pixel samples from non-edge pixel samples can be obtained.

4 Experiment results

We evaluate the proposed method on 32 astronomy images captured by the Cassini space detector. Four different measures are used to evaluate the performance. Our experiment is performed in Matlab that is run on the PC with 2.40 GHz Intel i5-4210 CPU and 8GB memory.

In the experiment, all the pixels of the test image are input into the above neural network. After the edge extraction result is obtained, the connected region of less than 10 pixels is removed, and the output result is compared with the labeled result to calculate the prediction accuracy rate and error rate.

4.1 Performance measures

There are four classification results: TP (True Positive), TN (True Negative), FP (False Positive), and FN (False Negative). TP refers to the number of pixels correctly classified as edges, TN refers to the number of pixels correctly classified as non-edges, FP refers to the number of pixels that incorrectly classified non-edges as edges, and FN refers to the number of pixels whose edges are incorrectly classified as non-edges. In the experiment, four measures including *Acc*, *Err*, *Sn*, and *Sp* are selected as the measurement criteria for edge extraction. *Acc* is the proportion of correct edge pixels to the total number of actual edge pixels. *Err* is the ratio of the number of edge pixels predicted to the total number of actual edge pixels. *Sn* is the sensitivity, also called the recall rate while *Sp* is the specificity and is also called the recall rate of negative cases. The description of measures is shown in Table 1.

Table 1. Performance measures of edge extraction.

Performance measures	Sensitivity(Sn)	Specificity(Sp)	Accuracy(Acc)	Error rate(Err)
Description	$TP/(TP+FN)$	$TN/(TN+FP)$	$(TP+TN)/(TP+FP+TN+FN)$	$FP/(TP+FN)$

4.2 Experimental results

In our experiment, 32 astronomy images captured by the Cassini space detector were used as the training set. There are over 80,000 pixel samples for training in all, and the ratio of edge samples to non-edge samples was approximately 1:4. The classifier training time is about 88.1719s.

After training, we use another seven Cassini astronomy images as test sets, putting them into the trained ELM model for edge extraction. The average accuracy reaches 0.9379 and the error rate is 0.1943 (as shown in Table 2). On the contrast, the effect of using traditional operators (Canny, Roberts, Sobel, Prewitt) is much worse, including too many internal edges and noise, as shown in Fig.3(b)-(e).

Fig. 3 shows the results of edge detection using different methods under different resolution Cassini astronomy images. Fig. 3 (a) shows two different resolution images. Fig. 3 (b)-(e) corresponds to the detection results of the Canny, Sobel, Roberts, and Prewitt operators, respectively. Fig. 3 (f) corresponds to our result. As is shown in Fig. 3, the edges obtained by the Canny, Roberts, and Prewitt operator cannot eliminate the false detection totally, and the connectivity of the edge of the observation object is different. However, our

ELM-based method can well extract the edge of the observation object, and can also get a better edge connectivity.

Table 2. Extraction results of our method.

Image No.	Acc	Err	Sn	Sp
01	0.9946	0.0285	0.9893	0.9999
02	0.8463	0.0625	0.8427	0.9999
03	0.9964	0.1912	0.9929	0.9994
04	0.9333	0.2500	0.9324	0.9995
05	0.9926	0.2731	0.9853	0.9990
06	0.9150	0.2881	0.9072	0.9991
07	0.9028	0.2978	0.8987	0.9991
Average	0.9401	0.1987	0.9335	0.9994
Max	0.9964	0.2978	0.9929	0.9999
Min	0.8463	0.0285	0.8427	0.9990

It can be proved that the method proposed in this paper has a better detection effect on the contour of the Cassini astronomy image with fewer false edges, and can achieve good results for astronomy images of different resolutions.

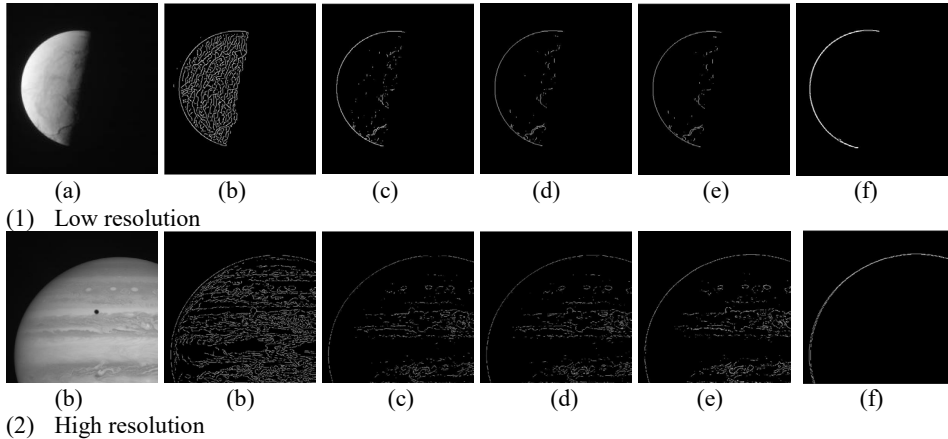


Fig. 3. Detected Edge by different methods on Cassini Astronomy image with low or high resolutions. (a) Original Image. (b) Canny. (c) Sobel. (d) Roberts. (e) Prewitt. (f) Our ELM.

5 Conclusion

In this paper, an edge detection method based on Extreme Learning Machine (ELM) for Cassini astronomy image is proposed. The edge extracted by traditional operators (such as Canny, Sobel, Roberts, etc.) contains too many inner details and noise. However, the experimental results show that the method proposed in this paper can extract the contour of the target objects in Cassini image more accurately (the average accuracy is about 0.9379), which takes an important step for the analysis of various physical properties of the observation object. What's more, this paper is also the first time to apply the Extreme Learning Machine to the contour extraction of Cassini astronomy image, which improves the accuracy of edge extraction and can provide reference for the other astronomy images.

Acknowledgments

This work was partly supported by the Joint Research Fund in Astronomy (U1431227) under cooperative agreement between the National Natural Science Foundation of China (NSFC) and Chinese Academy of Sciences (CAS), National Natural Science Foundation of China (Grant No. 11703008), Natural Science Foundation of Guangdong Province, China (Grant No. 2016A030313092, 2014A030313374) and the Fundamental Research Funds for the Central Universities.

References

1. C.A. Young, P.T. Gallagher, *Solar Image Analysis and Visualization* (Springer, New York, 247-59, 2008)
2. A. Liu, D. Zhou, L. Chen, M. Chen, *J. Syst. Eng. Electron*, **26**, 1291-300 (2015)
3. S.M. Saheba, T.K. Upadhyaya, R.K. Sharma. *IET Image Process*, **10**, 657-61 (2016)
4. T. Cornet, O. Bourgeois, S. L. Mouélic, S. Rodriguez, C. Sotin, J.W. Barnes, et al. *J. Geophys. Res-Planet*, **117** (2012)
5. S.A. Walker, J.S. Sanders, A.C. Fabian, *Mon. Not. R. Astron. Soc*, **461**, 684-97 (2016)
6. D. Marr, E. Hildreth, *Proceedings of the Royal Society of London Series B Biological Sciences*, **207**, 187-217 (1980)
7. J. Kittler, *Image Vision Comput.* **1**, 37-42 (1983)
8. L. Canny, *IEEE T. Pattern. Anal.* **8**, 679-98 (1986)
9. P. Dollár, Z. Tu, S. Belongie. *IEEE Conference on Computer Vision and Pattern Recognition*. 1964-71 (2006)
10. C. Farabet, C. Couprie, L. Najman, et al. *IEEE T. Pattern. Anal.* **35**, 1915-29 (2013)
11. P. Dollár, C.L. Zitnick, *IEEE T. Pattern. Anal.* **37**, 1558-70 (2015)
12. S. Xie, Z. Tu. *Int. J. Comput. Vision*, **125**, 3-18 (2017)
13. P. Viola, M. Jones. *Proc. of the IEEE Computer Vision and Pattern Recognition*. Cambridge, Britain, 511-18 (2001)
14. P. Viola, M.J. Jones. *Int. J. Comput. Vision*, **57**, 137-54 (2004)
15. R. Lienhart, J. Maydt. *Proceedings of IEEE International Conference on Image Processing*. Rochester, New York, USA, **9**, 900-03 (2002)
16. G.B. Huang, Q.Y. Zhu, C.K. Siew, *Neurocomputing*, **70**, 489-501 (2006)
17. J. Tang, C. Deng, G.B. Huang, B. Zhao. *IEEE Trans. Geosci. Remote Sens.* **53**, 1174-85 (2015)
18. M.D. McDonnell, M.D. Tissera, T. Vladusich., A.V. Schaik, J. Tapson. *PLOS ONE*, **10**, e0134254 (2015)
19. R. Minhas, A. Baradarani, S. Seifzadeh, et al. *Neurocomputing*, **73**, 1906-17 (2010)
20. R. Barandela, R.M. Valdovinos, J.S. Sanchez, et al. *LNCS, Springer Berlin Heidelberg*, **3138**, 806-14 (2004)
21. N.V. Chawla, K.W. Bowyer, L.O. Hall, et al. *J. Artif. Intell. Res.*, **16**, 321-57 (2002)