Automatic Detection of Craters in Planetary Images: An Embedded Framework Using Feature Selection and Boosting

Wei Ding[1], Tomasz F. Stepinski[2], Lourenco Bandeira[3], Ricardo Vilalta[4]
Youxi Wu[5], Zhenyu Lu[5], and Tianyu Cao[5]
University of Massachusetts Boston, Boston, Massachusetts[1]
Lunar and Planetary Institute, Houston, Texas[2]
Instituto Superior Tecnico, Lisbon, Portugal[3]
University of Houston, Houston, Texas[4]
University of Vermont, Burlington, Vermont[5]
ding@cs.umb.edu,tom@lpi.usra.edu,lpcbandeira@ist.utl.pt,vilalta@cs.uh.edu,
{youxiwu,zlu,tianyu.cao}@cems.uvm.edu

ABSTRACT

Identifying impact craters on planetary surfaces is one fundamental task in planetary science. In this paper, we present an embedded framework on auto-detection of craters, using feature selection and boosting strategies. The paradigm aims at building a universal and practical crater detector. This methodology addresses three issues that such a tool must possess: (i) it utilizes mathematical morphology to efficiently identify the regions of an image that can potentially contain craters; only those regions, defined as crater candidates, are the subjects of further processing; (ii) it selects Haar-like image texture features in combination with boosting ensemble supervised learning algorithms to accurately classify candidates into craters and non-craters; (iii) it uses transfer learning, at a minimum additional cost, to enable maintaining an accurate auto-detection of craters on new images, having morphology different from what has been captured by the original training set. All three aforementioned components of the detection methodology are discussed, and the entire framework is evaluated on a large test image of $37,500 \times 56,250\ m^2$ on Mars, showing heavily cratered Martian terrain characterized by nonuniform surface morphology. Our study demonstrates that this methodology provides a robust and practical tool for planetary science, in terms of both detection accuracy and efficiency.

Categories and Subject Descriptors

I.5.2 [**Design Methodology**]: [Classifier design and evaluation; Feature evaluation and selection; Pattern analysis]; I.5.4 [**Pattern Recognition**]: Applications—*Astronomy*

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

General Terms

Algorithms

Keywords

classification, feature selection, transfer learning, spatial data mining, planetary and space science

1. INTRODUCTION

Impact craters, the structures formed by collisions of meteoroids with planetary surfaces, are among the most studied geomorphic features in the solar system because they yield information about the past and present geological processes and provide the only tool for measuring relative ages of observed geologic formations [7, 20]. However, advances in surveying craters present in images gathered by planetary probes have not kept up with advances in collection of images at ever-higher spatial resolutions. As a result, there are "millions" of craters waiting to be identified in a deluge of high resolution planetary images but no means for their efficient identification and comprehensive analysis. Today, as in the past, craters are identified using manual inspection of images. As a result, comprehensive catalogs of craters are restricted to only large craters: $42,283$ Martian craters with diameters larger than 5 km [3], and $8,497$ named lunar craters with diameters larger than a few kilometers [1]. If left to manual surveys, the fraction of cataloged craters to the craters actually present in the available and forthcoming imagery data will continue to drop precipitously. Crater auto-detection techniques are needed, especially to catalog smaller craters that are most abundant. Surveying such craters is ill-suited for visual detection, due to their shear numbers, but well-suited for an automated technique. In fact, automating the process of small crater detection is the only practical solution to a comprehensive surveying of such craters.

Some challenges [14] of designing an accurate crater detection algorithm are as follows: (i) Craters, as a landform formation, lack strong common features distinguishing them from other landform formations. Their sizes differ by orders of magnitude. Their rims have often been eroded since their formation millions of years ago, resulting in shapes

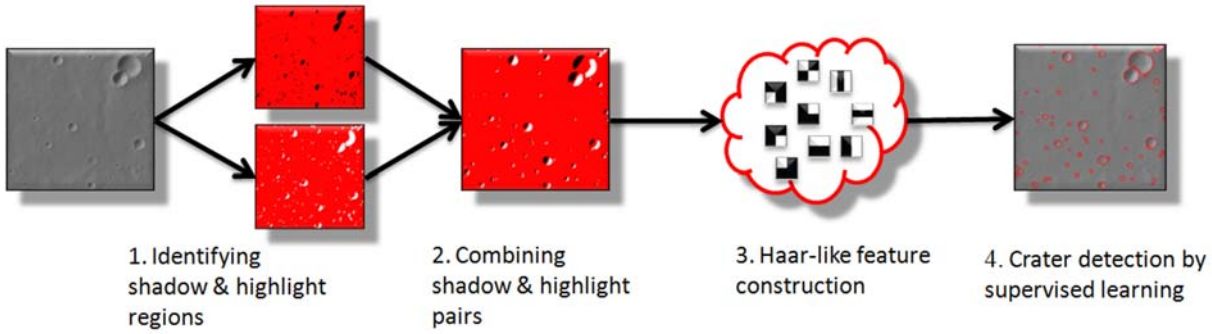


Figure 1: Diagram illustrating the crater-detection framework. (1) Crescent-like shadow and highlight regions are identified. (2) Shadow and highlight regions that can be matched are used to construct crater candidates. (3) Haar-like features are calculated using 9 kernels on crater candidates. (4) Craters are identified using supervised learning.

that depart significantly from circles. They frequently overlap, complicating the task of their separation from background. (ii) Planetary images are taken at different lighting conditions, at different resolutions, and their quality varies so that even morphologically identical craters may have different appearances in different images. (iii) Appearances of other similar landform formations exist in images. For example, volcanic cones and valleys fragments may resemble craters.

A robust and useful crater detection algorithm must address these challenges. From a design point of view such an algorithm has to successfully address the following issues: (i) How to efficiently identify crater candidates—regions in an image that have relatively high probability of containing a crater? (ii) Given a set of crater candidates, how to accurately classify them into crater and non-crater objects? An efficient approach in feature construction and selection and a well designed supervised learning approach are desirable to address this issue. (iii) Given a training set of crater candidates containing positive and negative examples (craters and non-craters), how to make a classifier applicable to other images, where candidates have a morphological character different from what is encapsulated by the training set? This is the scenario where training and testing instances are not drawn independently and identically from a same underlying distribution. Transfer learning is needed to address this issue while minimizing the overall cost of classification.

Previous research on crater detection algorithms [16, 10, 6, 2, 14, 23, 25, 5, 17] (also see [19] for a complete bibliography of research on auto-detection of craters) focused predominantly on addressing issue (ii). The problem of finding crater candidates has only recently been raised [22]; and the bulk of previous work relies on inefficient exhaustive search of the entire image. This may work for finding a small number of large craters in low resolution images, but not for finding a very large number of small craters in high resolution images. To the best of our knowledge, the problem of transfer learning in the context of auto-detection of craters was not previously addressed. This omission renders most existing approaches impractical for planetary research as the benefit of automation decreases significantly if new training sets need to be established for every new image or even for various segments of the same image.

This paper addresses all the three design issues in constructing a robust and practical crater detection algorithm, using an embedded framework with feature selection and boosting. The ultimate goal is to construct a robust and reliable crater auto-detection framework that can be widely adopted for planetary research. The flow chart indicating components of our method is shown in Fig. 1. The three key contributions are as follows:

- Utilizing mathematical morphology [21] for efficient identification of crater candidates. The key insight behind our method is that a crater can be recognized as a pair of crescent-like highlight and shadow regions in an image (see Fig. 2). The benefits of identifying crater candidates before the classification stage, rather than classifying pixel-based image blocks resulting from exhaustive search of the entire image are two-fold: (i) Significant computation time is reduced at the classification stage, and (ii) the number of false positives detections is reduced, as most of the image, including background, is removed from being classified.
- Using a combination of Haar-like image texture features [24] and a modified AdaBoost ensemble supervised learning algorithm. This approach yields a highly accurate classifier. As an alternative, we also evaluate the use of a simpler classifier (hereafter referred to as the "Naive" classifier) for distinguishing between craters and non craters.
- In order to minimize training for application of a crater detection algorithm to a heterogeneous planetary surface, we modify the basic boosting algorithm so it incorporates transfer learning to feature selection and classification.

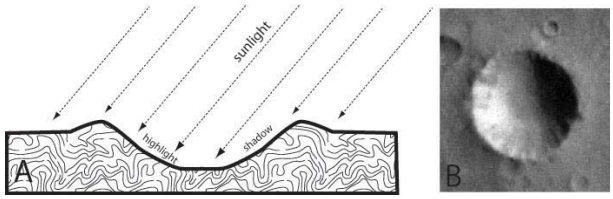


Figure 2: (A) Diagram explaining why an image of a crater consists of crescent-like highlight and shadow regions. (B) An image of an actual 1 km crater showing the highlight and shadow regions.

The entire method is evaluated on a large, high resolution

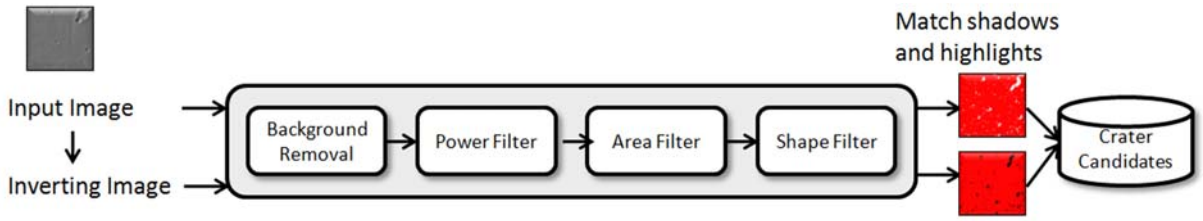


Figure 3: Diagram illustrating individual steps in constructing crater candidates

image of Martian surface ($37,500 \times 56,250\ m^2$) featuring spatial variability of crater morphology. Experimental results demonstrate robustness and good accuracy that validates our approach and makes it feasible to embed our algorithm into a processing pipeline for auto-creation of global, "million crater" catalogs of small craters on Mars, Mercury, and the Moon.

The rest of the paper is organized as follows. Section 2 provides a brief review on crater-detection methods. Sections 3 and 4.1 explain how we construct crater candidates and Haar-like texture-based features from those candidates. Sections 4.2 and 4.3 discuss the three supervised learning algorithms used for crater detection, and Section 5 presents the result of applying our methodology to find craters in the test image. Section 6 summarizes our work and discusses future directions.

2. RELATED WORK

Salamuniccar et al. provide a complete and exhaustive review of all previous research on crater detection algorithms [19]. All existing approaches to detecting craters in planetary images can be divided into two general categories: unsupervised approaches and supervised approaches.

The unsupervised methods identify crater rims in an image as circular or elliptical features [16, 10, 6, 2, 14]. In particular, the original image is preprocessed [16, 2, 14] to enhance the edges of rims, and the actual detection is achieved by means of the Hough Transform (HT) [11] or genetic algorithm [10]. Unsupervised methods have the advantage of being fully autonomous but they are generally less accurate than supervised methods.

The supervised methods [5, 23, 25] take advantage of domain knowledge in the form of labeled training sets that guide the classification algorithm. In [5, 23], a continuously scalable template-model technique is used to achieve detection. In [25], a number of algorithms are tested and the Support Vector Machine algorithm is shown to achieve the best rate of crater detection. More recent methods [14, 17] incorporate techniques originally developed [24] for the purpose of face detection. These methods concentrated on the classification component of crater detection and did not incorporate identification of crater candidates or transfer learning, as what has been designed and implemented in this paper.

3. CRATER CANDIDATES

In order to reduce the load on the classification module, we first identify crater candidates—parts of an image that contain crescent-like pairs of shadows and highlights. Identification of crater candidates is achieved using an image processing method based on mathematical morphology proposed by Urbach et al. on object detection in [22, 21]. Fig. 3 shows a flow diagram of the method used for identification of crater candidates. The highlight and shadow shapes are processed in parallel using inverted image to process the shadow shapes. The goal is to eliminate all the shapes that are not indicative of craters while keeping the highlight and shadow shapes. Background removal deletes shapes, such as mountains, that are too large to be part of the craters; the power filter removes shapes that lack sufficient contrast; the area filter removes shapes that are too small for reliable crater detection; the shape filter uses shape attributes that are invariant to translation, rotation, and scaling [21] to preserve or remove regions of an image exclusively on the basis of their shapes. Utilization of the shape filter, that requires only a singe parsing of an image, improves performance by a factor of 5 to 9 in comparison with other shape detection methods [21]. In the final step, the algorithm matches highlight and shadow regions so that each pair corresponds to a single crater candidate. This method does not have high enough accuracy to constitute a stand-alone crater detection technique, but is ideal for identification of crater candidates.

4. CLASSIFICATION

The classification stage uses supervised learning to distinguish (with high accuracy) between craters and non craters in the set of crater candidates. The objects of classification are image blocks containing crater candidates and the classification is performed on the basis of Haar-like image texture features.

4.1 Haar-like feature construction

We use image texture features reminiscent of Haar basis functions first proposed in [18] for detection of objects and later popularized by [24] in the context of face detection. These features can be thought of as image masks consisting of black and white sectors. The value of a feature is the difference between the sum of gray scale values in pixels located within the white sectors and the black sectors. We use nine types of features (all squares) as shown in Fig. 4. All features can be calculated very efficiently in one image scanning using the concept of integral image [24].

To represent a crater candidate in terms of Haar-like features, we first extract square image blocks around each crater candidate having size twice that of the candidate. For examples of image blocks and features positioned over them see Fig. 8. The underlying texture information of each crater candidate is encoded in the set of nine features, having various granularities and positioned at finely sampled locations. Thus an image containing a crater candidate and its immediate surrounding is described by thousands of texture features. Those features are not independent from each other and those over-complete features compensate the lim-

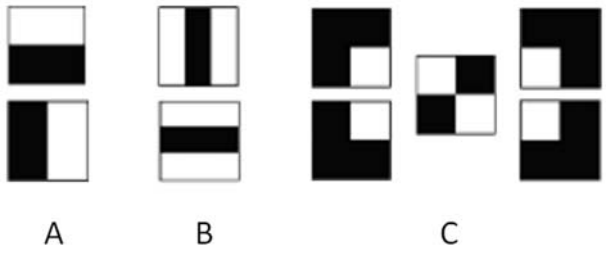


Figure 4: 9 rectangle features: (A) 2 two-rectangle features, (B) 2 three-rectangle features, (C) 5 four-rectangle features.

ited texture information a single rectangle feature can capture. If a single simple feature can be viewed as a weak learner, that is, only using this feature to classify a crater candidate, it is a natural choice to build a strong classifier out of thousands of weak learners, using the boosting ensemble method.

4.2 Classifiers: Boost and Naive

To classify crater candidates into craters and non craters on the basis of texture features, we have designed and implemented two supervised learning algorithms. These algorithms simultaneously select sub-set features necessarily for accurate classification and train the final ensemble classifier based on the supplied training set. The first is the Boost algorithm, a variant of the AdaBoost algorithm inspired by the methodology of face detection [24]. The second is the Naive algorithm—a drastic simplification of the Boost algorithm without boosting iterations. The two algorithms are described in Algorithms 1 and 2.

The Boost algorithm generates a sequence of weak classifiers $h_t(f)$ and combine them through a weighted approach to build a strong classifier $H(\widehat{x})$:

$$H(\widehat{x}) = \sum_{t=1}^{T} \alpha_t h_t(f), \tag{1}$$

where T is the number of iterations, $t = 1, \ldots, T$, $\widehat{x} =< f_1, \ldots, f_N >$ is the feature vector that describes a crater candidate; f, $f \in \{f_1, \ldots, f_N\}$, is the single feature used to construct a weak classifier $h_t(f)$, and α_t is the weight of hypothesis $h_t(f)$. The Boost algorithm (See Algorithm 1) iteratively selects one feature at a time and stops when reaching T iterations; note that $T << N$. Each iteration selects one best feature and consists of three core steps:

1. Weak Classifier Learning: The construction of weak classifier $h_t(f)$ on a single feature f at iteration t is straightforward. Given examples $(\widehat{x_1}, y_1), \ldots, \widehat{x_n}, y_n)$ where $y_i = 0, 1$ $(i = 1, \ldots, n)$ for non-crater and crater examples respectively, a weak classifier $h_t(f)$, consists of a feature f, a threshold θ, and a polarity p indicating the direction of the inequality:

$$h_t(f) = \begin{cases} 1 & if\ f(\widehat{x}) < p\theta \\ 0 & otherwise \end{cases} \tag{2}$$

   A weak leaner $h_t(f)$ can be essentially viewed as a decision stump, a single-node decision tree.

2. Feature Selection: Calculate the weighted error sum of each weak classifier and select the best learner (a.k.a. the best feature) that produces the minimum error.

3. Weight Updating: Update weights using the same method proposed in AdaBoost [9]—increase the weights of incorrectly classified examples and decrease the weights of correctly classified examples. The incorrectly classified examples will have more chances of being chosen in the next iteration when calculating the weighted error sum in step 2. Hence, the next selected feature concentrates more on the mistakes made by the earlier features. The key advantage of the Boost algorithm is that the weights encode the classification results of the previous features and this information is used to select the next best feature.

The number of craters usually is less than the number of non-craters. The initial weight of each training instances is designed to cope with imbalance data by using different group average weights in the positive and negative classes, respectively.

Algorithm 1 Boost: A boosting algorithm for feature selection and classification

Require: .

(1) Given crater candidates $(\widehat{x_1}, y_1), \ldots, (\widehat{x_n}, y_n)$ where $y_i = 0, 1, i = 1, \ldots, n$ for non-crater and crater examples respectively.

(2) Initialize weights $w_i = \frac{1}{2m}$ if $y_i = 0$, $w_i = \frac{1}{2l}$ if $y_i = 1$, where m and l are the number of non-crater and crater examples respectively.

1: **for** $t = 1 \ldots T$ **do**

2: Normalize the weight, $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, i = 1, \ldots, n$ so that w_t is a probability distribution.

3: Select the best weak classifier with respect to the weighted error $\epsilon_t = argmin_{f,p,\theta} \sum_i w_i |h(\widehat{x_i}, f, p, \theta) - y_i|$, For each feature, f, train a classifier h, which is restricted to using a single feature.

4: Define $h_t(\widehat{x}) = h(\widehat{x}, f_t, p_t, \theta_t)$, where $\widehat{x}, f_t, p_t, \theta_t$ are the minimizers of ϵ_t.

5: Update the weights: $w_{t+1,i} = w_{t,i}\beta_t^{1-e_i}, i = 1, \ldots, n$ where $e_i = 0$ if a crater candidate x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

6: The final strong classifier is: $h(\widehat{x}) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$ where $\alpha_t = ln\frac{1}{\beta_t}$ and μ is a user-defined threshold.

7: **end for**

In order to reduce the computational cost of the Boost algorithm, we design a simplified greedy version of the algorithm and call it the Naive algorithm (see Algorithm 2). The Naive classifier uses the same *weak classifier learning* step and selects the top T best features using the weighted error sum in the step of *feature selection* as a criterion without any further iterations on *weight updating*.

4.2.1 Time Complexity Analysis

The time complexity of the Boost algorithm is $O(TNn)$, where n is the number of training examples, N is the number of total features, and T is the number of boosting iterations.

In particular, each feature produces n weak classifiers, based on each feature value for every training example according to the threshold θ; N features produce Nn classifiers; it takes $O(Nn)$ time to find the weak classifier that produces the minimum error; and it takes $O(TNn)$ time to select the top T features after T boosting iterations.

The time complexity of the Naive algorithm is $O(Nn)$ as no boosting iterations are performed. Interestingly, the Naive classifier performs well in some circumstances during our real-world case study (see Section 5).

Algorithm 2 Naive: A naive greedy algorithm for feature selection and classification

Require: .

(1) Given crater candidates $(\widehat{x_1}, y_1), \ldots, (\widehat{x_n}, y_n)$ where $y_i = 0, 1, i = 1, \ldots, n$ for non-crater and crater examples respectively.

(2) Initialize weights $w_i = \frac{1}{2m}$ if $y_i = 0$, $w_i = \frac{1}{2l}$ if $y_i = 1$, where m and l are the number of non-crater and crater examples respectively.

1: Normalize the weight, $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, i = 1, \ldots, n$ so that w_t is a probability distribution.

2: Select the best t ($t = 1, \ldots, T$) weak classifiers with respect to the weighted error $\epsilon_t = \sum_i w_i |h(\widehat{x_i}, f, p, \theta) - y_i|$, For each feature, f, train a classifier h, which is restricted to using a single feature.

3: Define $h_t(\widehat{x}) = h(\widehat{x}, f_t, p_t, \theta_t)$ where $\widehat{x}, f_t, p_t, \theta_t$ are the minimizers of ϵ_t, and $t = 1, \ldots, T$.

4: $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$.

5: The final strong classifier is:
$$h(\widehat{x}) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=1}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$
where $\alpha_t = ln \frac{1}{\beta_t}$ and μ is a user-defined threshold.

4.3 Transfer learning

Boost and Naive assume that both training and testing instances are drawn independently and identically from a same underlying distribution. What if training and test instances are from different distributions? We have designed a transfer learning based algorithm, inspired by [8], which is capable of transferring knowledge from the old training data to the new test data. We refer to it as the TL algorithm. The TL algorithm (see Algorithm 3) has the same three steps as the Boost algorithm, but the *weight updating* step is different as it attempts to transfer knowledge from the original training set to the new test data. The Boost algorithm assumes that both, training and test instances are drawn independently and identically from an underlying distribution and it is not expected to perform well if the test data has a different distribution from the training data; this is because the critical set of features that best serves to distinguish craters in the training set may not be the same as that in the test set.

We denote the previous original training data as the diff-distribution training data; here we are uncertain about the similarity and usefulness of this data for the new task. And we denote the additional small portion of labeled test data, which is representative of the new set of crater candidates, as the same-distribution training data. During the training process, we apply the Boost algorithm to the same-distribution training data to build a model; the weights of misclassified examples are increased during the next iteration while the weights of correctly classified examples are decreased. The key component is that we transfer knowledge from the old training data to the new test data by modifying the weights of misclassified examples from the diff-distribution training data. Those misclassified examples are considered as the ones that are dissimilar to the same-distribution examples and should be de-emphasized. Accordingly, we decrease (not increase) the weights of those examples in order to weaken their impact. The weight-changing mechanism selects good examples (similar to the labeled test data) from the old training data to compensate the insufficient training examples in the same-distribution data.

The difference between the TL algorithm and the existing algorithm TrAdaBoost [8] is that we use embedded approach in feature selection. In our method, we select the best feature in each iteration while constructing a strong classifier sequentially. The key contribution of the algorithm is that some features contribute more in the new test data and should be transferred and emphasized, while some features provide less or no contributions at all and thus should be de-emphasized. The sub-set features best discriminates craters and non-craters in old training set is not necessarily the same sub-set features in a new unseen test set. The change of weight factor $\beta = \frac{1}{1+\sqrt{2ln\frac{n}{T}}}$ for misclassified examples from diff-distribution and the threshold voting $\sum_{t=\lceil \frac{T}{2} \rceil}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=\lceil \frac{T}{2} \rceil}^{T} \alpha_t$ in the final strong classifier are to assure that the average training loss on the diff-distribution converges to zero [8, 9].

5. EXPERIMENTAL RESULTS

5.1 Test image

We have selected a portion of the High Resolution Stereo Camera (HRSC) nadir panchromatic image h0905 [12], taken by the Mars Express spacecraft, to serve as the test set. As illustrated in Fig. 10, the selected image has the resolution of 12.5 meters/pixel and the size of $3,000$ by $4,500$ pixels ($37,500 \times 56,250\ m^2$). A domain expert manually marked $\sim 3,500$ craters in this image to be used as the ground truth to which the results of auto-detection are compared. The image represents a significant challenge to automatic crater detection algorithms because it covers terrain having spatially variable morphology and because its contrast is rather poor (this is most noticeable when the image is inspected at a small spatial scale). We divide the image into three sections denoted as the west region, the central region, and the east region (see Fig. 10). The central region is characterized by surface morphology that is distinct from the rest of the image. The west and east regions have similar morphology but the west region is much more heavily cratered than the east region.

5.2 Training Set Construction

In the first stage of our method, we identify 13,075 crater candidates in the image using the pipeline depicted in Fig. 3. The data set is imbalanced as the majority objects are non-crater candidates. 1,089 Haar-like features are constructed using the 9 rectangle features described in Fig. 4. The training set for the Boost and Naive algorithms consists of

	# of candidates	Precision			Recall			F1		
		Boost	Naïve	TL	Boost	Naïve	TL	Boost	Naïve	TL
West Region	6708	0.897	0.889	**0.911**	0.791	**0.821**	0.779	0.841	**0.853**	0.840
Central Region	2935	0.805	0.772	**0.902**	0.783	**0.816**	0.769	0.794	0.794	**0.830**
East Region	3432	0.915	0.867	**0.954**	0.809	**0.855**	0.843	0.892	0.861	**0.895**
All Regions	13075	0.881	0.856	**0.919**	0.807	**0.827**	0.791	0.842	0.841	**0.851**

Figure 5: Performance results of the Boost, Naive, and TL algorithms; parameter values: Boost, 150 features selected and $\mu = 0.525$; Naive, 150 features selected and $\mu = 0.675$; TL, 150 features selected and $\mu = 0.500$.

Algorithm 3 TL: A boosting algorithm using transfer learning for feature selection and classification

Require:

(1) Given a training set that includes crater candidates $(\widehat{x_1}, y_1), \ldots, (\widehat{x_{n_d}}, y_{n_d}), (\widehat{x_{n_d+1}}, y_{n_d+1}), \ldots, (\widehat{x_{n_d+n_s}}, y_{n_d+n_s})$, where $y_i = 0, 1, i = 1, \ldots, n_d, n_d + 1, \ldots, n_d + n_s$ for non-crater and crater examples respectively.

This training set has n_d diff-distribution examples $(1, \ldots, n_d)$ and n_s same-distribution examples $(n_d + 1, \ldots, n_d + n_s)$, and $n = n_d + n_s$.

(2) Initialize weights $w_i = \frac{1}{2m}$ if $y_i = 0$, $w_i = \frac{1}{2l}$ if $y_i = 1$, where m and l are the number of non-crater and crater examples respectively

1: **for** $t = 1 \ldots T$ **do**

2: Normalize the weight, $w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^{n} w_{t,j}}, i = 1, \ldots, n$ so that w_t is a probability distribution.

3: Select the best weak classifier with respect to the weighted error
$\epsilon_t = argmin_{f,p,\theta} \sum_i w_i |h(\widehat{x_i}, f, p, \theta) - y_i|$, $i = n_d + 1, \ldots, n_d + n_s$
For each feature, f, train a classifier h in same-distribution data, which is restricted to using a single feature.

4: Define $h_t(\widehat{x}) = h(\widehat{x}, f_t, p_t, \theta_t)$ where $\widehat{x}, f_t, p_t, \theta_t$ are the minimizers of ϵ_t

5: Update the weights:
$w_{t+1,i} = w_{t,i}\beta_t^{-e_i}$, if $n_d + 1 \leq i \leq n_d + n_s$ (increase the weights for the same-distribution)
$w_{t+1,i} = w_{t,i}\beta^{e_i}$, if $1 \leq i \leq n_d$ (decrease the weights for the diff-distribution)
where $e_i = 0$ if example x_i is classified correctly, $e_i = 1$ otherwise, and $\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$, $\beta = \frac{1}{1+\sqrt{2ln\frac{n}{T}}}$

6: The final strong classifier is:
$$h(\widehat{x}) = \begin{cases} 1 & \sum_{t=\lceil \frac{T}{2} \rceil}^{T} \alpha_t h_t(\widehat{x}) \geq \mu \sum_{t=\lceil \frac{T}{2} \rceil}^{T} \alpha_t \\ 0 & otherwise \end{cases}$$
where $\alpha_t = ln\frac{1}{\beta_t}$ and μ is a user-defined threshold.

7: **end for**

204 true craters and 292 non-crater examples selected randomly from amongst crater candidates located in the northern half of the east region. Thus, the training set uses only 3.75% of the total data set. Note that we have purposely restricted the locations of examples in a training set to a specific sector of the image in order to mimic actual planetary research; it is likely that in current studies such craters are identified in a specific region and are in need of identification by a supervised learning algorithm in the rest of the image. For the TL algorithm, we have constructed an additional training set (same-distribution set) consisting of 253 crater candidates (102 true craters and 153 non-craters) selected from random locations throughout the entire image. The ratio between the false and true examples in the same-distribution data is proportional to that in the diff-distribution data ($\frac{153}{102} \approx \frac{292}{204}$). The original training set consisting of 496 examples from the northeastern section of the image serves as the diff-distribution set.

5.3 Supervised Learning

The table shown in Fig. 5 summarizes the performance results of crater detection by the three algorithms: Boost, Naive, and TL. The ground truth of the entire image serves as an external criterion to evaluate the performance of the three algorithms on the unseen test set. Of the three algorithms, the number of features used to construct a strong classifier and the values of the threshold μ are selected to maximize the performance of each classifier.

The candidate data has imbalanced class distribution and the successful detection of true craters is more significant than the detection of non-craters. Hence we use recall ($r = \frac{TP}{TP+FN}$) and precision ($p = \frac{TP}{TP+FP}$) and F1 as the evaluation metrics. TP stands for the number of true positive detections (detected craters that are actual craters), FP stands for the number of false positive detections (detected craters that are not), and FN stands for the number of false negative "detections" (non-detection of real craters). F1 measures the harmonic mean between precision and recall $\frac{2}{\frac{1}{r}+\frac{1}{p}}$. The values of precision, recall, and F1 are listed, and the best performance of each measure is highlighted in bold. A precision score of 1.0 means that every object classified as a crater is indeed a crater but says nothing about the number of craters that are not recognized by classifiers as such. A recall score of 1.0 means that every true crater is classified as such but says nothing about how many other landforms were incorrectly classified as craters. An F1 score

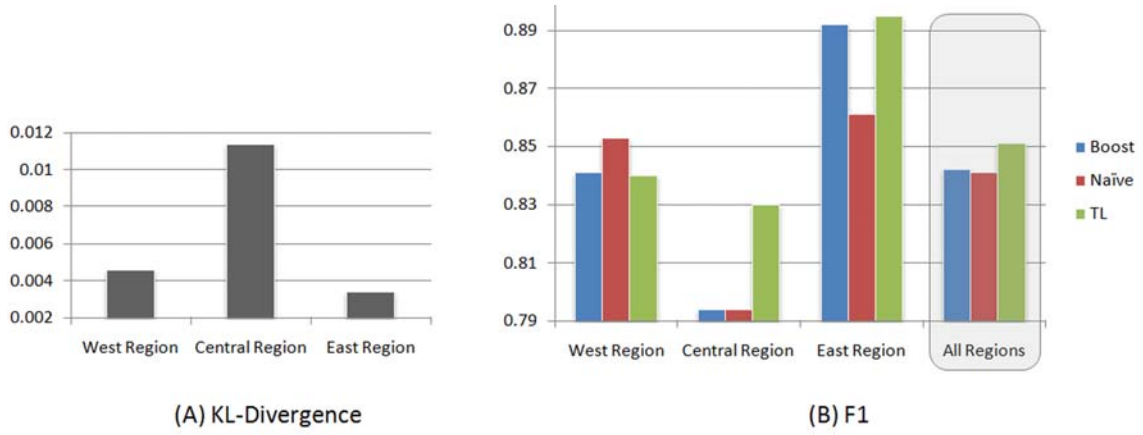


Figure 6: (A) Kullback-Leibler divergence measures between the set of feature vectors in the original training set and the sets of feature vectors in the west, central, and east regions. (B) Graphical illustration of F1 scores of the three algorithms. (Best viewed in color.)

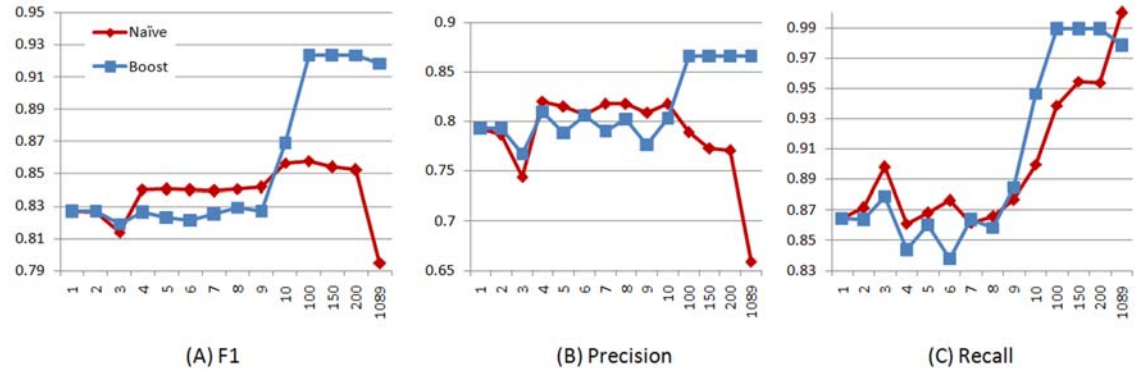


Figure 7: Boost versus Naive. X-axis: number of features selected; Y-axis: Performance scores.

of 1.0 means that all the existing craters are correctly identified and all the objects classified as craters are true craters.

The TL classifier yields the best precision in all regions and the Naive classifier yields the worst precision in all regions. On the other hand, the Naive classifier has the highest recall in all regions whereas the TL classifier has the lowest value of recall, except in the east region, where the Boost classier has the lowest value of recall. Overall, the TL classifier has the highest value of F1 in all regions except the west region where the Naive classifier has the highest value of F1.

5.4 Comparative Study

We also tested three representative algorithms for the purpose of comparative study: AdaBoost [9] with C4.5 as the base leaner is used as an example of boosting algorithms, SVM [4, 13] with a linear kernel is used as an example of kernel-based learning algorithms, and TrAdaBoost [8] with C4.5 as the base leaner is used an example of transfer learning algorithms. Using all the 1089 features, the F1 score of SVM on all regions is 0.202, AdaBoost is 0.302, TrAdaBoost is slightly better than 0.4. The huge performance gain by the three algorithms (Boost, Naive, and TL) has been achieved by the effective integration of the feature-selection with supervised learning.

5.5 Transfer Learning

In order to better understand the results of three proposed algorithms, it is useful to assess dissimilarity between the set of features vectors in the original training set and those in the west, central, and east regions. Fig. 6A shows such dissimilarity as measured by the Kullback-Leibler divergence [15]; Fig. 6B plots the F1 scores graphically of 3 regions. Clearly, the central region is most dissimilar to the training set, whereas the east region is the most similar (since the training set was selected from the northeastern portion of the image). This is why the TL classifier performs best (relatively to other classifiers) in the central region. It is expected that the TL classifier would have the least advantage in the east region, as it is the region best characterized by the training set, but the results shows that the TL classifier has the smallest gain (if any) in the west region. This can be explained by the fact that the west region has a similar character to the east region, but is much more heavily cratered, so in fact, relatively fewer additional training samples come from these regions resulting in no sufficient information gain to be exploited by the TL classifier.

The Naive classifier performs surprisingly well considering its simple nature and low computational cost. We took an in-depth look into the performance of the Boost and Naive classifiers on the northeastern section of the image contain-

Figure 8: Top 6 features selected by Naive, Boost, and TL, respectively

ing 1406 crater candidates of which 496 constitute a training set for both algorithms. Fig. 7 shows the precision, recall, and F1 for these classifiers as a function of the number of features selected to construct a strong classifier. The Boost classifier clearly outperforms the Naive classifier on F1 and precision measures if more than 100 features are selected. However, the recall measures of the two classifiers remain comparable regardless of the number of selected features. Thus, the Boost classifier is superior to the Naive classifier on crater candidates that closely resemble those in the training set, but that disadvantage decreases and/or disappears when classifying crater candidates that are less similar to those in the training set. We link the relatively small advantage (or lack of advantage) of the Boost classifier over the naive classifier to the peculiarity of image texture features in the context of crater detection. Top features (weak classifiers) are actually quite strong performers by themselves capable of achieving an F1 score as high as 0.81. These features limit the advantage of the boosting algorithm that works best with an ensemble of weak classifiers.

5.6 Feature Selection

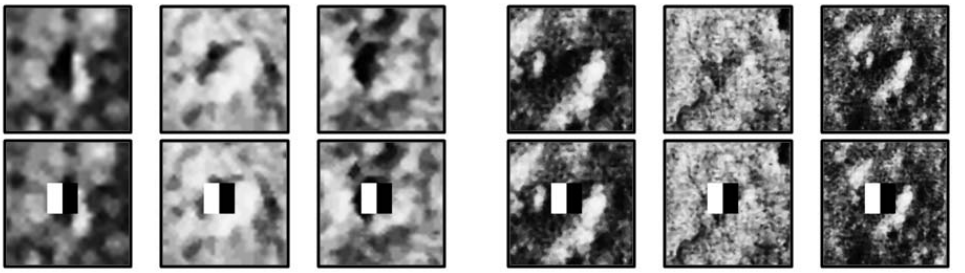
It is instructive to compare top features (weak classifiers) selected by each of the three classification algorithms (Naive, Boost, and TL). Fig. 8 shows six top features selected by each algorithm. The top two features selected by the three algorithms concentrate on the transition between the shadow and the highlight which best define the characteristics of a cater, but there are significant differences between other selected top features. Features selected by the Naive algorithm are relatively strong by themselves. Most of them utilize the transition between the shadow and the highlight to distinguish craters from no craters. While the next best feature selected by the Boost algorithm always attempts to correct mistakes done by the previous feature. Fig. 9 illustrates how the second best feature selected by the Boost algorithm corrects the mistakes by the first best feature, and we can observe that this feature performs well on candidates with shifted shadow regions. Not all top features selected by the TL algorithm utilize the transition between shadows and highlights, but rather crater rims. This indicates the new test data has different characteristics on crater edges.

Fig. 10 displays the results of the TL algorithm, using top 150 features and the threshold $\mu = 0.500$. Notice that the large craters $\geq 5000-$meter in diameter are intentionally not detected as we set the parameters of our algorithm to target small sub-kilometer craters (large craters on Mars have already been identified manually [3]).

Our methodology is relatively fast. Using a machine with 2 CPUs, 3.16 GHZ, 8GB RAM it took 360 seconds to generate crater candidates in the test image. It took, on average, 5 seconds to classify 13,075 candidates using a pre-trained classifier. The total computing time is around $\frac{365}{3000*4500} = 27$ microseconds per pixel.

6. CONCLUSIONS AND FUTURE WORK

The aim of this paper is to present a robust and practical framework for auto-detection of small craters in high resolution images of planetary surfaces. This is one of the most challenging problems in planetary science—effective and automatic crater detection from extremely large orbiter images. The framework uses an innovative method that integrates improved techniques on three key components: identification of crater candidates, embedding feature selection with supervised classification, and transfer learning. First, we have demonstrated that our method identifies craters with high accuracy. The test site is an HRSC image of Martian scene that presents a heterogeneous region of $37,500 \times 56,250$ m^2 and craters in various forms which are challenging for detection using regular algorithms. Our method can achieve an F1 score of 0.851; an algorithm with such performance can definitively be used in planetary research. Second, we have demonstrated that a consistently accurate detection can be achieved with a minimum cost through transfer learning. Without transfer learning the performance of our algorithms (Boost and Naive) decreases in the central region of the image where surface morphology differs as characterized by the training set. However, using the TL algorithm partially restores the level of performance. Third, we noticed that the Naive algorithm can perform well



(A) 1st row: True craters misclassified by the 1st best feature; (B) 2nd row: True craters overlaid by the 2nd best feature.

(B) 1st row: Non-crater candidates misclassified by the 1st best feature; (B) 2nd row: Non-crater candidates overlaid by the 2nd best feature.

Figure 9: The second best feature selected by the Boost algorithm successfully classified 6 misclassified examples using the first best feature.

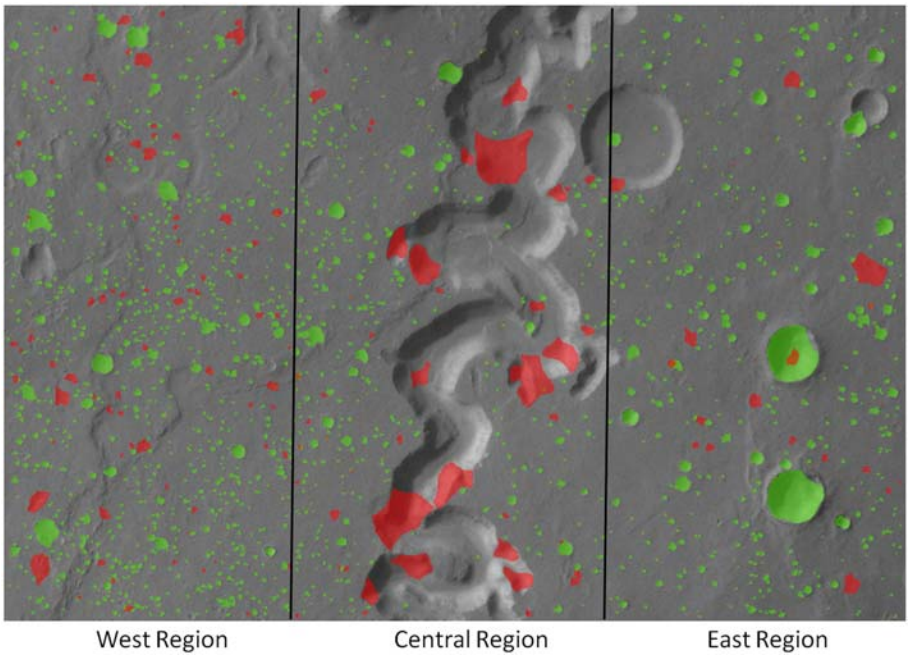


Figure 10: Craters (<= 5000-meter in diameter) detected in a $37,500 \times 56,250\ m^2$ test image. (Best viewed in color.) Green: True detections, Red: False detections.

in the context of crater detection for a fraction of the cost of the Boost algorithm.

We contend that the robustness and practicality of our methodology make its utilization in planetary research likely. If adopted, our method has great potential to produce surveys of small craters over entire surfaces of planets, thus revolutionizing certain aspects of planetary science. Our future research will address means of efficient selection of additional training samples for construction of the same-distribution for transfer learning. The goal is to intelligently select samples that exemplify differences between the existing training sets and new candidate sets.

7. ACKNOWLEDGMENTS

The work is partially supported by NASA grant NNX09AK86G. The authors would like to thank Dr. Xindong Wu of the University of Vermont for informative discussions and valuable feedback and comments, and Dr. Qiang Yang of the Hong Kong University of Science and Technology for constructive advice on transfer learning.

8. REFERENCES

[1] L. E. Andersson and E. A. Whitaker. NASA catalog of Lunar nomenclature. In *NASA Reference Publication 1097*, 1982.

[2] L. Bandeira, J. Saraiva, and P. Pina. Impact crater recognition on Mars based on a probability volume created by template matching. In *IEEE Transactions on Geoscience and Remote Sensing*, pages 4008 – 4015, 2007.

[3] N. G. Barlow. Crater size-frequency distributions and a revised Martian relative chronology. *Icarus*, 75:285–305, 1988.

[4] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *COLT '92: Proceedings of the 5th annual workshop on Computational learning theory*, pages 144–152, Pittsburgh, Pennsylvania, United States, 1992.

[5] M. C. Burl, T. Stough, W. Colwell, E. B. Bierhaus, W. J. Merline, and C. Chapman. Automated detection of craters and other geological features. In *Int. Symp. Artif. Intell. Robot. and Autom. Space*, Montreal, QC, Canada, 2001.

[6] Y. Cheng, A. E. Johnson, L. H. Matthies, and C. F. Olson. Optical landmark detection for spacecraft navigation. In *the 13th Annual AAS/AIAA Space Flight Mechanics Meeting*, AAS 03-224, pages 1785–1803, Puerto Rico, Puerto Rico 2002.

[7] Crater Analysis Techniques Working Group. Standard techniques for presentation and analysis of crater size-frequency data. *ICARUS*, 37:467–474, February 1979.

[8] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 193–200, Corvalis, Oregon, 2007.

[9] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational Learning Theory: Eurocolt*, pages 23–37. Springer-Verlag, 1995.

[10] R. Honda, Y. Iijima, and O. Konishi. Mining of topographic feature from heterogeneous imagery and its application to lunar craters. In *Progress in Discovery Science, Final Report of the Japanese Discovery Science Project*, page 395407, London, UK, 2002. Springer-Verlag.

[11] P. C. Hough V. Method and means for recognizing complex patterns. United States Patent, December 1962.

[12] HRSC Data Browser. http://europlanet.dlr.de/node/index.php?id=209, 2009.

[13] T. Joachims. *Learning to Classify Text using Support Vector Machines*. PhD thesis, Kluwer Academic Publishers, 2002.

[14] J. Kim, J.-P. Muller, S. Van Gasselt, J. Morley, and G. Neukum. Automated crater detection: a new tool for Mars cartography and chronology. *Photogrammetric Engineering and Remote Sensing*, 71(10):1205–1217, October 2005.

[15] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[16] B. Leroy, G. Medioni, and E. J. L. Matthies. Crater detection for autonomous landing on asteroids. *Image and Vision Computing*, 19:787–792, September 2001.

[17] R. Martins, P. Pina, J. Marques, and M. Silveira. Crater detection by a boosting approach. *IEEE Geoscience and Remote Sensing Letters*, 6:127–131, 2009.

[18] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *6th International Conference on Computer Vision*, pages 555 – 562, January 1998.

[19] G. Salamuniccar and S. Loncaric. Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on MOLA data. *Advances in Space Research*, 42(1):6–19, 2007.

[20] K. L. Tanaka. The stratigraphy of Mars. *Journal of Geophysical Research*, 91:E139–E158, November 1986.

[21] E. R. Urbach, J. B. T. M. Roerdink, and M. H. F. Wilkinson. Connected shape-size pattern spectra for rotation and scale-invariant classification of gray-scale images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:272–285, 2007.

[22] E. R. Urbach and T. F. Stepinski. Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, 57:880–887, 2009.

[23] T. Vinogradova, M. Burl, and E. Mjolsness. Training of a crater detection algorithm for Mars crater imagery. In *IEEE Aerospace Conference Proceedings*, volume 7, pages 7–3201–7–3211, 2002.

[24] P. Viola and M. J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2 2004.

[25] P. Wetzler, R. Honda, B. Enke, W. Merline, C. Chapman, and M. Burl. Learning to detect small impact craters. In *Seventh IEEE Workshops on Application of Computer Vision*, volume 1, pages 178–184, Breckenridge, CO, 2005.