

Adaptive Selective Learning for Automatic Identification of Sub-Kilometer Craters

Siyi Liu^a, Wei Ding^a, Feng Gao^b, Tomasz F. Stepinski^c

^a{sliu,ding}@cs.umb.edu, Department of Computer Science, University of Massachusetts Boston, Boston, MA, 02125.

^bfeng.gao-1@nasa.gov, Hydrospheric and Biospheric Sciences Laboratory, NASA Goddard Space Flight Center, Greenbelt, MD20771

^cstepintz@uc.edu, Department of Geography, University of Cincinnati, Cincinnati, OH 45221

Abstract

Counting craters is a fundamental task of planetary science, because it provides the only tool for measuring relative ages of planetary surfaces. However, advances in surveying craters present in data gathered by planetary probes have not kept up with advances in data collection. It becomes extremely challenging to automatically count a very large number of small, sub-kilometer size craters in a deluge of high resolution planetary images. In this paper, we combine active learning with semi-supervised learning to build an adaptive learning system to automatically detect craters from high resolution panchromatic planetary images. We propose an adaptive selective algorithm to iteratively enrich an original small training set, using unlabeled test set without additional human labeling effort, to detect craters from a large volume of images. We propose three strategies to improve detection accuracy by integrating classification with exploration on unlabeled samples. The Majority Vote Strategy is used to automatically obtain class labels by exploiting unlabelled samples. The De-Mixed Strategy is used on instance filtering to obtain reliable samples. The Active Stability Strategy is used to obtain an appropriate class distribution in the constructed training set by detecting unstable classes. By using those three strategies, we actively select test instances from test images into an existing small initial training set while rebuilding the classifier in the mean time. Our proposed algorithms are empirically evaluated on a large high resolution Martian image, exhibiting a heavily cratered Martian terrain characterized by heterogeneous surface morphology. The experimental results demonstrate that the proposed approach achieves a higher accuracy than other existing approaches to a large extent.

Keywords-*pattern recognition, active learning, semi-supervised learning, crater detection, Mars*

1 Introduction

Impact craters are approximately circular depressions in the surface of a planet or other solid body in the Solar System, formed by the hyper-velocity impact of smaller bodies with the surface. Craters 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 [Tanaka, 1986]. Presently, all crater surveys are done by means of visual inspection of images which are restricted to only large craters; for example, 42,283 Martian craters with diameters larger than 5 km have been cataloged in [Barlow, 1988]. As a result, there are millions of sub-kilometer craters waiting to be identified in high resolution planetary images. As a matter of fact, the size distribution of craters conforms to the power-law as the large craters are rare and small craters are abundant [Tanaka, 1986]. Counts of significant number of craters, especially small craters, must be collected from spatially extended regions in order to accumulate sufficient number of samples for accurate statistics. Geologic stratigraphy based on manually collected databases has coarse spatial resolutions. Finer spatial resolutions of the stratigraphy can only be obtained from statistics of smaller craters, and the only viable means to obtain spatially comprehensive databases of small sub-kilometer craters is through automating the process of crater detection.

It becomes extremely challenging to automatically count a very large number of small, sub-kilometer size craters in a deluge of high resolution planetary images. Identification of craters in remotely sensed images can be considered as a special case of object detection in images—an important task in computer vision exemplified by a popular task of face detection. However, craters have characteristics unlike most objects traditionally subjected to automated identification in images, because they are numerous, have large range of sizes, and they continuously merge into a background. Craters lack specific features that can reliably discriminate them from other objects, or collection of objects, also present on planetary surfaces, including volcanic cones and valley fragments resembling craters.

We apply supervised learning to crater detection in this paper. Many factors impact the detection rate, including feature construction and selection, training set construction, and classifier induction. In this paper, we focus on the problem of training set construction. It is impractical to construct a large training set sampled from a large pool of planetary images because a domain expert can only manually label a small image in a particular location. We propose the Adaptive Selective Algorithm to iteratively enrich an original small training set using unlabeled test set, without additional human labeling effort, to detect craters from a large volume of images. We propose three strategies on three different tasks to improve detection accuracy by integrating classification with exploration on unknown test samples. The Majority Vote Strategy is used to automatically obtain class labels by exploiting unlabeled samples. The De-Mixed Strategy is used on instance filtering to obtain reliable samples. The Active Stability Strategy is used to obtain accurate class distribution in the constructed training set

by detecting unstable classes. By using those three strategies, we actively select unlabeled test instances from test images into existing small initial training set and rebuild the classifier. Our proposed algorithm is empirically evaluated on a large high resolution Martian image, containing 3,500 sub-kilometer craters, which are manually labeled by a domain expert. The study site presents a challenging case for any crater detection task as it exhibits a heavily cratered Martian terrain of $37,500 \times 56,260 \text{ m}^2$, characterized by heterogeneous surface morphology. The experiment results demonstrate that the proposed approach achieves a higher accuracy than other existing approaches to a large extent.

The paper is organized as follows. Section 2 discusses the related work. Section 3 explains how we construct crater candidates and extract Haar-like texture-based features from those candidates. Section 4 discusses the proposed Adaptive Selective Learning Algorithm for crater detection, and Section 5 presents the result of applying our methodology to find craters in the test site on Mars. Section 6 summarizes our work and discusses future directions.

2 Related Work

Existing approaches of detecting craters from planetary images can be divided into two general categories of unsupervised and supervised methods, and the latter case requires inputs of a domain expert to construct an initial training set.

The unsupervised methods rely on image processing techniques to identify crater rims in an image as circular or elliptical features [Leroy et al., 2001; Honda et al., 2003; Cheng et al., 2003; Barata et al., 2004; Kim et al., 2005]. The original image is preprocessed to enhance the edges of the rims, and the actual detection is achieved by means of the Hough Transform (HT) [Hough, 1962], genetic algorithms [Honda et al., 2003], or the radial consistency algorithm [Earl et al., 2005] that identifies regions of rotational symmetry. Specifically, in [Leroy et al., 2001; Honda et al., 2003; Cheng et al., 2003], the preprocessing consists of applying an edge-detecting algorithm; in [Barata et al., 2004], it consists of calculating texture measures; and in [Kim et al.2005], it consists of a combination of edge detection, texture measurement, and edge direction analysis. The performance of unsupervised methods on crater detection is usually worse than that of supervised methods.

The supervised methods [Burl et al., 2001; Vinogradova et al., 2002; Plesko et al., 2004; Wetzler et al., 2005] use machine learning concepts to build a classifier model from a training set to detect craters. In a learning phase, the training set of images, containing craters labeled by a domain expert, is fed into a learning algorithm. In the detection phase, the previously induced classification model detects craters in a new, unlabeled set of images. In [Burl et al., 2001] and [Vinogradova et al., 2002], a continuously scalable template-model technique is used to achieve detection. In [Wetzler et al., 2005], a number of algorithms are tested and the Support Vector Machine (SVM) algorithm is shown to achieve the best rate of crater detection. In [Plesko et al., 2004], genetic

programming is used to generate a population of random-detection algorithms whose performance is iteratively improved using a training set as selection criteria. Most recently, advances in face detection research are incorporated into crater detection techniques. In [Kim et al. 2005], the combination of edge detection, template matching, and neural network-based false positive recognition scheme is used for detecting craters on Mars. In [Martins et al., 2008] a boosting algorithm, originally developed by [Viola and Jones, 2004] in the context of face detection, is adopted for identification of craters on Mars. In [Ding, 2010], a boosting transfer learning algorithm is used for crater detection.

To the best of our knowledge, neither active learning nor semi-supervised learning has been studied in the field of crater detection. Our method combines the active learning and semi-supervised learning to construct different training set according to different test set to achieve higher accuracy.

Active learning [Dasgupta and Hsu, 2008; Ertekin et al., 2007; Settles, 2009] has gained much attention recently. In an active learning setting, a classifier is first trained from an initial small training set, and the classifier is used to classify an unlabeled test set. Then, it selects instances from the unlabeled data set, asks a domain expert to label those selected instances, and adds those instances into the training set. There are three scenarios in which active learning algorithms may pose queries: the membership query synthesis [Augluin, 1988], in which, the learner may request labels for any unlabeled instances including the queries that the learner generates; the stream-based selective sampling [Cohn et al., 1990, 1994], in which, the unlabeled instances are sampled from the actual distribution, and then the learner decides whether to query or discard it, and the pool-based sampling [Lewis and Gale, 1994], in which, instances are selected from a large pool of unlabeled data set. However, the active learning requires a human annotator to label all instances that the algorithm selects. It is still quite time-consuming and expensive thus impractical in crater detection from remotely sensed images. The Self-Training [Yarowsky, 1995] in semi-supervised learning is an approach for building the training set from the unlabeled data set automatically. Self-Training assumes that a classifier’s prediction, at least the high confidence ones, tend to be correct. It uses unlabeled data without additional human effort and only selects those with high confidence. It may increase the accuracy on data in small labeled size problems [Culp and Michailidis, 2007; Haffari and Sarkar, 2007].

The proposed Adaptive Selective Learning Algorithm in this paper is different from active learning. Active learning still needs additional human effort on labeling, while we obtain the class labels automatically when expanding the original training set. Our method is also different from Self-Training. Self-Training selects the instances with high confidences while our method achieve better detection accuracy using the Major Vote Strategy, De-Mixed Strategy, and Active Stability Strategy to select the instances from the unlabeled data set. Our method combines active learning using the pool-based sampling scenario and semi-supervised learning to construct the training set dynamically according to a different unlabeled test set.

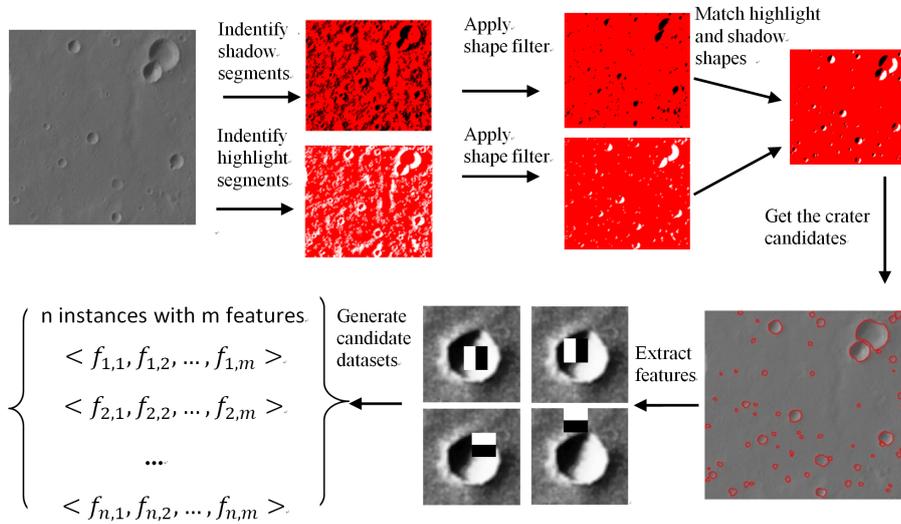


Figure 1: Pipeline for constructing a crater candidate dataset

3 Constructing Crater Candidate Datasets

Figure 1 shows a diagram illustrating a pipeline for generating crater candidate data, which will later be the inputs for the proposed learning algorithm.

The input of the pipeline is a panchromatic image which contains many highlight and shadow regions. The pipeline processes highlight and shadow regions in parallel using inverted image to process the shadow regions. First, we identify shadow and highlight segments from the input image and apply the shape filter to remove shapes that likely do not belong to craters. Then we match the highlight and shadow regions to get the locations that are possible to be craters, namely, crater candidates. Criteria to match highlight and shadow regions to build a crater candidate include the distance, the similarity, and the circularity of the those two regions. Detailed discussion is provided in [Urbach and Stepinski, 2009]. After that, we extract features from each crater candidates.

We define a crater candidate is the region that is indicative for craters, where highlight and shadow regions can be matched. We construct a feature vector for each candidate using Haar-like features [Papageorgiou et al. 1998]. These features are simple texture features which are calculated using image masks consisting only black and white sectors. The value of a feature is the difference between the sum of gray pixel values located within the black sector and the white sector of an image mask. Figure 2 shows nine image masks used in our case study. The first five masks focus on capturing diagonal texture gradient changes, while the remaining four masks on horizontal or vertical textures. In order to detect craters in different diameters, each sampled image block is normalized



Figure 2: Nine types of masks used for Haar-like feature extraction in our real world case study

to a same image block. Each of these nine kinds of image masks probe the normalized image block in different scales, with a step in the ratio of the mask size. The total number of generated Haar-like features is determined by the number of image masks, the number of scales used for each image mask, the size of steps when scanning the masks across the normalized image block.

This pipeline is different from our previous work in [Urbach and Stepinski, 2009] with the following improvements. The pipeline uses thousands of Haar-like image gradient texture features to represent crater candidates, while Urbach and Stepinski [2009] uses several mathematical morphology features on geometric shapes. Using such geometric features to distinguish true craters from other objects present in the set of candidates yields less than optimal classification results [Urbach and Stepinski, 2009]. Recent study [Martins et al., 2008] has demonstrated that image gradient texture features are effective discriminants between craters and non-crater objects. In addition, we employ semi-supervise with adaptive selective learning to construct a training set according to different test set, while the work in [Urbach and Stepinski, 2009] simply use a prefixed manually built training set on crater detection. As to the crater dataset, it is impractical and expensive to construct a large comprehensive training set. The proposed Adaptive Selective Learning Algorithm automatically labels the instances without additional human effort to enrich an original small training set.

4 Adaptive Selective Learning

4.1 Problem Explanation

In this paper, we aim to design and implement a robust classification learning algorithm that is capable of dealing with a large test set when the initial training set is small. In the case of crater detection, it is impractical to ask domain experts to label a large spectrum of craters in many images. Inevitably, the critical set of features that distinguishes craters from non-craters in the training set, which is only generated from a small set of image, may not well represent those in a test set.

We present a new adaptive selective learning approach to actively select instances from the test set to the training set to enrich the original small training set. While adding the instances from the unlabeled data set to the training set, we must deal with the following three problems:

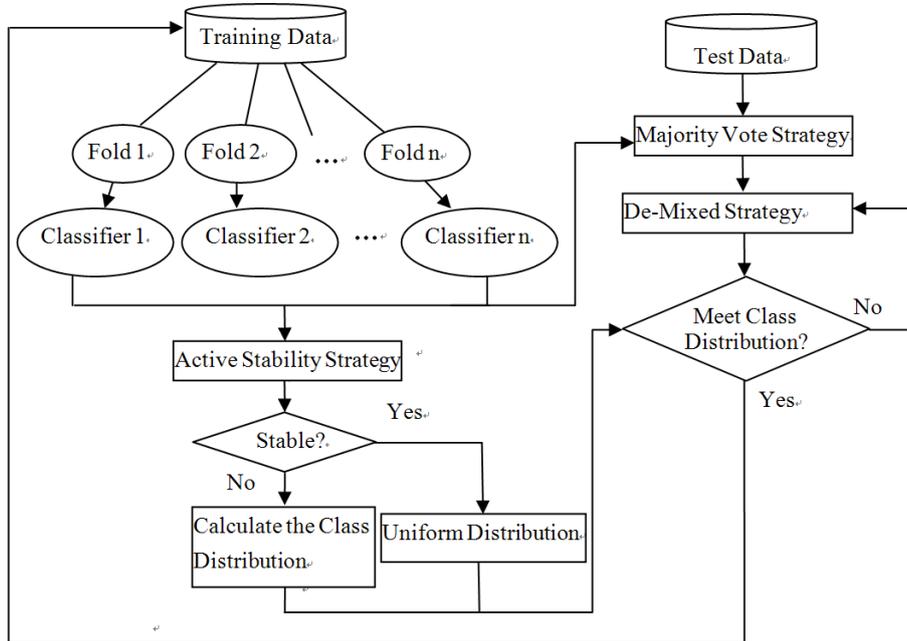


Figure 3: The flowchart of the Adaptive Selective Learning Algorithm

Class Label Acquisition : How can we automatically decide class labels while adding unlabeled instances to the training set? Our method does not need a human annotator to label instance on the test set. Instead, the learning algorithm will label the unlabeled instances as craters or non-craters.

Instance Filtering : Which instances should we select from the unlabeled data set? That is, our algorithm needs to judge whether an instance in the new test set can help induce a better classifier for the test data. If the instance has a positive impact on classifier induction, its class label will be determined automatically and the instance will be added into the training set. Otherwise, the instance will not be used.

Active Class Selection : How many instances should we add with respect to each class, iteratively? In our crater detection case study, we have two classes, craters and non-craters. We need to dynamically decide the class ratio between the crater class and non-craters class while expanding the original training set.

4.2 Adaptive Selective Learning

The Adaptive Selective Learning algorithm is designed to solve the problems of class label acquisition, instance filtering and active class selection. Figure 3 depicts the whole process of the algorithm. The algorithm starts with a small training set and iteratively adds new selected test instances from the test set to the training set. Specially, our algorithm first partitions the initial training set to n folds and uses a supervised learning algorithm (e.g. SVM) to build n classifiers from the n folds. The algorithm applies the n classifiers to test set and produces n labels for each test instance. Then we use the Majority Vote Strategy (see section 4.3) to obtain the class labels to address the class label acquisition problem. After labeling the instances in the test set, we use the De-Mixed Strategy (see section 4.4) to select instances to address the instance filtering problem. At the same time, on the training set, we use the Active Stability Strategy (see section 4.5) to test whether class distribution in the current training set is stable to address the active class selection problem, if it is stable, we use uniform distribution, if not, we calculate the desired class distribution. Then, the algorithm selects the instances until it satisfies the expected class distribution. The algorithm adds those newly selected test instances into the training set and perform the whole process again until no more qualifying new test instances can be found.

4.3 Majority Vote Strategy

The Majority Vote Strategy is designed to solve the class label acquisition problem: how can we automatically decide the class labels while adding the unlabeled instances to the training set?

Let T be the training set, $T = \{(\vec{x}_i, y_i)\}_{i=1}^t$, where $\vec{x}_i = \langle f_{i,1}, f_{i,2}, \dots, f_{i,m} \rangle$ is the feature vector of instance \vec{x}_i in the training set, y_i is the class label of the instance \vec{x}_i . $y_i \in \{0, 1\}$ for non-crater and crater instances, respectively. t is the number of instance in the training set. Let U be the test dataset, $U = \{\vec{x}_j\}_{j=1}^u$, $\vec{x}_j = \langle f_{j,1}, f_{j,2}, \dots, f_{j,m} \rangle$ be the feature vector of instance \vec{x}_j in the test set, u is the number of instances in the test set. Let $p_{j,f}$ be the predicted label of \vec{x}_j using the classifier C_f , where $p_{j,f} \in \{0, 1\}$ is for non-crater and crater instances, respectively.

The Adaptive Selective Learning Algorithm builds n classifiers $\{C_f\}_{f=1}^n$. The Majority Vote Strategy assigns a class label to a new test instance if the majority votes of a crater candidate indicating it belongs to this class:

$$\frac{\sum_{f=1}^n p_{j,f}}{n} > \varphi_1, j = 1 \dots \mu \quad (4.1)$$

$$\frac{\sum_{f=1}^n p_{j,f}}{n} < \varphi_2, j = 1 \dots \mu \quad (4.2)$$

φ_1 and φ_2 are the user-defined thresholds, and $\varphi_1 \geq \varphi_2$. If the value of $\frac{\sum_{f=1}^n p_{j,f}}{n}$ is greater than φ_1 , then we classify x_j as a crater, if it is smaller than

φ_2 , then we classify x_j as a non-crater. If $\varphi_2 \leq \frac{\sum_{f=1}^n p_{j,f}}{n} \leq \varphi_1$, then we do not label this candidate, because the n classifiers are uncertain about the class label of this candidate. And the Majority Vote Strategy will not add this instance to the training set. This strategy carefully selects new instances on which the classifiers have high confidence and avoids those instances on which the existing training set cannot consensus. The higher value of φ_1/φ_2 , less but more strong crater/non-crater examples will be selected. The lower value of φ_1/φ_2 , more but less strong crater/non-crater examples will be selected.

4.4 De-Mixed Strategy

The De-Mixed strategy is designed to solve the instance filtering problem: Which instances should we select from the unlabeled data set?

The intuition behind this strategy is that we want to build a training set that is well separable by the decision boundary of the classifier and has as few noisy data as possible. If a crater candidate \vec{x}_j , successfully labeled by the Majority Vote Strategy, is closer to the instances in the training set in the same class label and is farther from the instances in the training set in the different class label, then $(\vec{x}_j, P(\vec{x}_j))$, is added to the training set, where $P(\vec{x}_j)$ is the class label obtained by the Majority Vote Strategy. Otherwise, the instance will not be added into the training set.

$$\frac{\sum_{i=1}^{n_c} dist(\vec{x}_j[c], \vec{x}_i[c])}{n_c} - \frac{\sum_{i=1}^{n_{c'}} dist(\vec{x}_j[c], \vec{x}_i[c'])}{n_{c'}} < \mu, j = 1, \dots, \nu \quad (4.3)$$

where $\vec{x}_j[c]$ is the instance in the test set with predicted label c , $\vec{x}_i[c]$ is the instance in the training set with the same label c . $\vec{x}_i[c']$ is the instance in the training set with label c' , and $c \neq c'$. μ is the user-defined threshold. ν is the number of instance successfully labeled by the Majority Vote Strategy; note that ν is less than μ , the total number of instances in the test set. $dist(\vec{x}_j[c], \vec{x}_i[c])$ calculates the distance between an instance in the test set and an instance in the training set. $\sum_{i=1}^{n_c} dist(\vec{x}_j[c], \vec{x}_i[c])$ obtains the sum of the distance between the instance $\vec{x}_j[c]$ with class label c in the test set and all the instances with label c in the training set. $\frac{\sum_{i=1}^{n_c} dist(\vec{x}_j[c], \vec{x}_i[c])}{n_c}$ obtains the average distance between the instance $\vec{x}_j[c]$ with class label c and all the instances with label c in the training set. The explanation for $\frac{\sum_{i=1}^{n_{c'}} dist(\vec{x}_j[c], \vec{x}_i[c'])}{n_{c'}}$ is the same, but class c is different from class c' . If $\mu = 0$, it means if the average distance between the same class is smaller than the average distance between the different classes, then we add $(\vec{x}_j, P(\vec{x}_j))$ to the training set. If $\mu \neq 0$, it means if the average distance between the same class is smaller than the average distance between the different classes plus μ , then the algorithm adds $(\vec{x}_j, P(\vec{x}_j))$ to the training set with a certain tolerance on noisy data.

4.5 Active Stability Strategy

The Active Stability Strategy is designed to solve the third problem: How many instances should we add regarding each class?

The idea behind this strategy is that the instances whose predicated class labels changed in this iteration compared to the last iteration are near volatile boundaries. Thus, we assess which classes are near volatile boundaries and sample more instances from those unstable classes in the test set.

The Active Stability Strategy is used only on the training set to obtain the sampling distribution. The strategy uses the n classifiers to predict the instances in the training set which composed by the initial training set and new instances selected from the test set by the Majority-Vote and De-Mixed Strategies. For a training example, if the new generated class label is different from the previous class label which is generated by the previous n classifiers using the training in last iteration, then it indicates that we need to sample from these unstable classes. We use equation (4.4) to obtain the sampling distribution.

$$Num(c) = \frac{\frac{unstable[c]}{n_c}}{\sum_{i=0}^{|classes|} \frac{unstable[i]}{n_c}} * \rho \quad (4.4)$$

where $Num(c)$ is the number of class c need to be added, $unstable[c]$ is the number of instances whose classification changes, n_c is the number of class c in the training set. We divide $unstable[c]$ by n_c to avoid small classes from being ignored and large classes from being over-emphasized. $|classes|$ is the number of classes. ρ is the total number of instances to be added in the round.

[Lomasky et al., 2007] proposes an active class selection algorithm named Redistricting which iteratively builds training sets using the labeled data. The major differences between Redistricting and the Active Stability Strategy are: first, Redistricting builds the training set using the labeled data, while the Active Stability Strategy builds the training set iteratively using the unlabeled data. Second, Redistricting does not consider which instance should be added, it just samples the instances from the labeled data randomly, while we select the high quality instances from the unlabeled data set actively using the Majority Vote Strategy and the De-Mixed Strategy.

4.6 Pseudocode of Adaptive Selective Algorithm

The pseudocode of the Adaptive Selective Algorithm is shown in Algorithm 1.

Algorithm 1 Adaptive Selective Algorithm

Input: (1) Training set $T = \{(\vec{x}_i, y_i)\}_{i=1}^t$, test set $U = \{\vec{x}_j\}_{j=1}^\mu$.
 (2) ρ , number of instances to be added in round r , user defined thresholds

φ_1, φ_2 and μ .

1 Initially, let $T_1 = T = \{(\vec{x}_i, y_i)\}_{i=1}^t, U = \{\vec{x}_j\}_{j=1}^\mu$ and $r = 1$.

```

2   Divide  $T_1$  into  $n$  stratified folds  $T_{1,1}, T_{1,2}, \dots, T_{1,n}$ 
3   For  $f = 1$  to  $n$  do
4       Build Classifier  $C_f$  from  $\{T_1 - T_{1,f}\}$ ,  $C_{f'}$  from  $T_{1,f}$ 
5       For all  $\vec{x}_i$  in  $T_{1,f}$  do label  $\vec{x}_i$  with  $P_r(\vec{x}_i)$  with  $C_f$  end For
6       For all  $\vec{x}_j$  in  $U$ , do label  $\vec{x}_j$  as  $p_{j,f}$  with  $C_{f'}$  end For
7   end For
8   If  $\frac{\sum_{f=1}^n p_{j,f}}{n} > \varphi_1, j = 1, \dots, \mu$ 
9       do label  $\vec{x}_j = 1$  (1 is for crater)
10  end If
11  If  $\frac{\sum_{f=1}^n p_{j,f}}{n} < \varphi_2, j = 1, \dots, \mu$ 
12      do label  $\vec{x}_j = 0$  (0 is for non-crater)
13  end If
14  While no instance can be added in  $T_{add}$ 
15      If  $r = 2$  then
16           $T_{add} =$  random sample size of  $\rho$ 
17      else
18          Compute  $Num(c) = \frac{unstable[c]}{\sum_{i=0}^{|classes|} \frac{n_c \cdot unstable[i]}{n_c}} * \rho$ 
19          Initialize  $counts(c), c \in \{0, 1\}$ 
20          For all  $x_j$ 
21              If  $\frac{\sum_{i=1}^{n_c} dist(\vec{x}_j[c], \vec{x}_i[c])}{n_c} - \frac{\sum_{i=1}^{n_{c'}} dist(\vec{x}_j[c], \vec{x}_i[c'])}{n_{c'}} < \mu$ 
22                  If  $counts(c) < Num(c)$ 
23                       $y_i = P(\vec{x}_j)$ 
24                      add  $(\vec{x}_j, y_j)$  to  $T_{add}$ 
25                       $counts(c) ++$ ;
26                  end If
27              end If
28          end For
29           $T_r = T_{r-1} + T_{add}$ 
30          Initialize  $unstable[c], c \in \{0, 1\}$ 
31          Divide  $T_{add}$  into  $n$  stratified folds  $T_{add,1}, T_{add,2}, \dots, T_{add,n}$ 
32          For  $f=1$  to  $n$  do
33               $T_{r,f} = T_{r-1,f} \cup T_{add,f}$ 
34              Build the Classifier  $C_f$  form  $\{T_r - T_{r,f}\}$ 
35              For all  $\vec{x}_i$  in  $T_{r,f}$  do label  $\vec{x}_i$  as  $p_r(\vec{x}_i)$  with  $C_f$  end For
36              If  $P_r(\vec{x}_i) \neq p_{r-1}(\vec{x}_i)$  unstable  $[y_i] ++$  end If
37          end For
38           $r ++$ 
39  end While

```

We first discuss how Active Class Selection is used in the Adaptive Selective Learning Algorithm. We begin with a Cross Validation [Kohavi,1995] with n folds over T_1 , the initial training set. We obtain a prediction for each $\vec{x}_i \in T_1$. In the second round, we collect T_2 of size ρ . We next perform a Cross Validation

over all of the data in the training set and create a classifier for each fold. Note that on subsequent iterations, we keep the data from T_{r-1} in the same folds, and stratify only the newly generated data T_{add} into the existing folds. For each fold, we compare the classification results of $P_r(\vec{x}_i)$ and $P_{r-1}(\vec{x}_i)$ on each instance $\vec{x}_i \in T_1$. If the labels are different, then the counter for the class specified by y_i , $unstable[y_i]$, is incremented. We conclude by generating predictions of the new batch of data T_{add} and the increment r . After the second round we add instances using the formula (4.4), where c is a class from the set of all classes in the dataset.

We explain the proposed Adaptive Selective Learning Algorithm in detail. Steps 1 to 2 initialize the training set and the test set. Steps 3 to 7 use Cross Validation to label instances in the training set, use each fold to build a classifier and label instances in the test set. Steps 8 to 13 use the Majority Vote strategy to obtain the class labels. Steps 15 to 38 use the Active Stability Strategy. Steps 20 to 21 check whether each instance satisfies the criteria of the De-Mixed Strategy. Steps 32 to 37 are used to get the $unstable[c]$. we add the newly generated folds to each fold in the last round $r - 1$ and build new classifier from $\{T_r - T_{r,f}\}$ and then compare the predicted label in this round and the previous round, if it is different. Then $unstable[y_i]$ is increased, y_i is the real class label of instance x_i if it is from the initial training set; otherwise y_i is the predicted label obtained by the Majority Vote Strategy. The stopping criterion is that no more instances can be added in T_{add} , that is, the instances in the test set which have not been chosen before do not satisfy all the three strategies, thus no more qualified instances can be added into the training set and the calculation stops.

5 Experiment Setup and Experimental Results

5.1 Test Image

We have selected a portion of the High Resolution Stereo Camera (HRSC) nadir panchromatic image h0905 [HRSC, 2010], taken by the Mars Express spacecraft, to serve as the case study site for crater detection. As illustrated in Figure 4, the selected image has the resolution of 12.5 meters/pixel and the size of 13,500,000 (3,000 by 4,500) pixels. A domain expert manually marked 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. It covers terrain having spatially variable morphology and 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. 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.

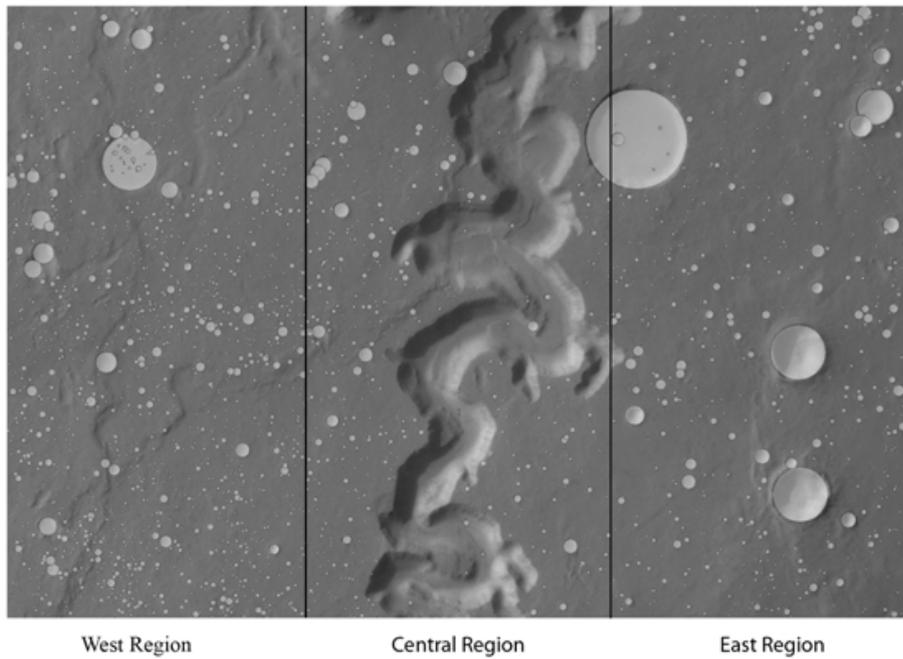


Figure 4: Site for the case study, located in the Xanthe Terra, centered on Nandedi Vallis and covers mostly Noachian terrain on Mars, from the image HRSC nadir panchromatic h0905, resolution of 12.5m/pixel. The site's north-south extension is 40 km, east-west extension is nearly 60 km and it is divided into 3 regions. The 3,500 craters in the manually established ground truth catalog are shown lighter gray circles with sizes proportional to the diameters of the craters. Vertical lines indicate boundaries between the west, central, and east regions.

5.2 Experimental Results

5.2.1 Experimental setup

We first test the sensitivity of the parameters chosen for the three proposed strategies. Figure 5 shows the experimental results on the φ_1 , φ_2 in the Majority Vote Strategy, μ in the De-Mixed Strategy, and ρ in the Active Stability Strategy. The accuracy improves with the value of φ_1 increases. This is reasonable, because the higher value that the classifiers get from the class label, the more confident that the label could be correct. While the accuracy decrease with the increase of φ_2 , this is due to the fact that more mislabeled samples are added with the increase of φ_2 . The accuracy first increases then later decreases with the increase of μ . This is because that the farther of μ is from 0, either in negative or positive values, the more mixed samples are allowed to be added into the training set. The accuracy is not sensitive to the ρ threshold in the Active Stability Strategy, because the learning process will be performed in many iterations, the accuracy will not change too much if the total number of instances selected from the test set are about the same.

We identify 12,542 crater candidates in the image using the pipeline depicted in Figure 1. Each crater candidate image block is normalized to a standard scale of 48 pixels. Each of the nine kinds of image masks probes the normalized image block in four different scales of 12 pixels, 24 pixels, 36 pixels, and 48 pixels, with a step of a third of the mask size (meaning $2/3$ overlap), returning 121 scale independent feature values. We totally extract 1,089 Haar-like features using nine types of masks illustrated in Figure 2 as the feature vectors to represent each crater candidate. Then, we use the feature selection algorithm described in [Ding, 2010] to select top 10 features from the 1,089 features and only those 10 best features are used in the experiments. The training set consists of instances selected randomly from amongst crater candidates located in the northern half of the east region. We set thresholds $\varphi_1 = \varphi_2 = 0.5$ for the Majority Vote Strategy, threshold $\mu = 0$ for the De-Mixed Strategy and ρ is set to 80 for each iteration for the Active Stability Strategy based on empirical observation. The training set consists of 204 positive instances and 39 negative instances.

5.2.2 Detecting the Difference between the Training Set and the Test Set

In order to better understand the results of our approach, it is useful to assess dissimilarity between the set of features vectors in the original training set and the sets of feature vectors in the west, central, and east regions. We have calculated the KL-divergence [Kullback and Leibler, 1951] between the training set and the test set. Formula 5.1 is the equation to calculate KL-divergence with two variables. In words, it is the average of the logarithmic difference between the probabilities P and Q, where the average is taken using the probabilities P. if the quantity $0 \times \log 0$ appears in the formula, it is interpreted as zero.

$$D_{KL}(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)} \quad (5.1)$$

Apparently, the KL-divergence is not symmetric. The KL-divergence from the west region, the central region and the east region to the training set is 0.65, 0.91, and 0.42. The smaller the value of the KL divergence is, the similar two images are. The central region has the largest divergence. The reason why the east region has smallest divergence is that the training set comes from a part of the east region. The KL-divergence indicates a robust algorithm should be able to achieves high accuracy in all different regions of this challenging case study site.

5.2.3 Experimental Results

Sub-kilometer craters marked by a domain expert are served as the ground truth in our performance evaluation process. The Adaptive Selective Learning Algorithm classifies crater candidates as craters and non-craters. The number of ground truths covered by the craters detected by the algorithm is TP (True Positives). The number of crater candidates that are not craters but mistakenly classified as craters is FP (False Positives). The number of crater candidates that are not craters and correctly classified as non-craters is TN (True Negatives). The number of crater candidates that are craters but failed to be identified as craters is FN (False Negatives). We compare accuracy ($accuracy = \frac{TP+TN}{TP+TN+FP+FN}$) between the Adaptive Selective Learning Algorithm and other three algorithms of random sampling, Redistricting and Self-Training.

- Random Sampling: it is a widely used approach for training data selection. It produces subset of the data which has a distribution similar to the original test data set by randomly sample each instance independently. This subset of data are added to the training set, and the class labels of those new instances are obtained by a classifier trained on the original training set. This approach is the most common way to get the class label in a traditional semi-supervised learning method.
- Redistricting: The original method was proposed in [Lomasky et al., 2007] and it used the idea of assessing the boundary classes. In our experiments, we implemented the core method of redistricting, that is, it assesses the unstable class and decides the class ratio to be achieved. To assure a fair comparison, the redistricting implemented in our paper does not use the true labels provided by a domain expert. Instead, we obtain the class labels by a classifier trained on the training set which is a typical way in a semi-supervised learning method.
- Self-Training: it is proposed in [Yarowsky, 1995] and it adds new instances with high confidence from the test set to the training set.

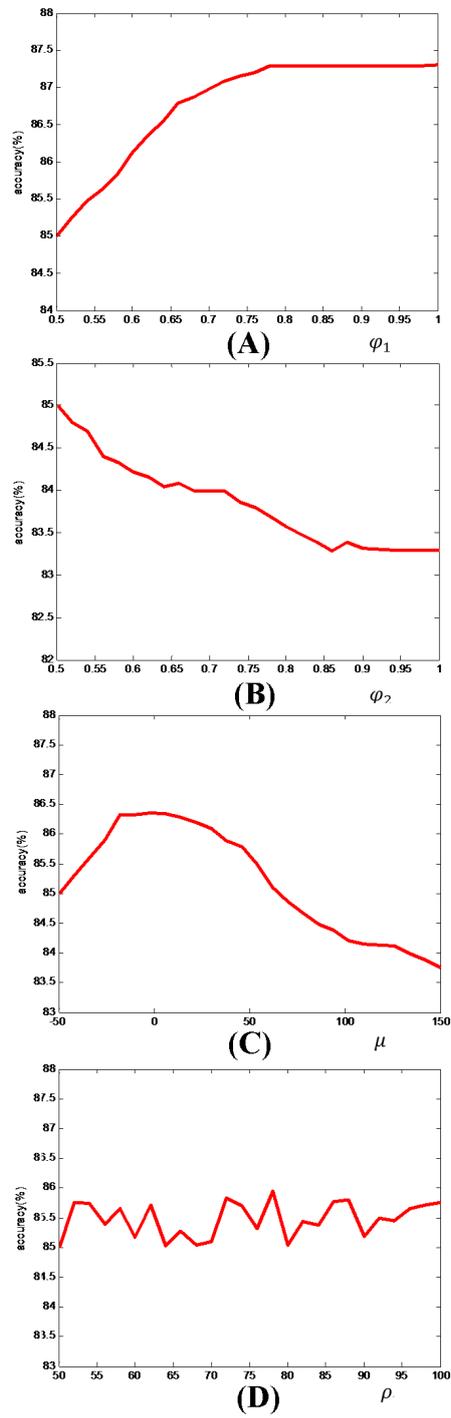


Figure 5: Experimental results of different threshold parameter values of $\phi_1, \phi_2, \mu, \rho$.

Figure 5 shows the accuracy using Adaptive Selective Learning, Random Sampling, Redistricting and Self-Training. The ground truth of the entire image serves as an external criterion to evaluate the performance of the four algorithms on the test set. Of the four algorithms, the base classifier used is LIBSVM [Chang, 2001], a SVM classifier using the radial basis function kernel (normalized Euclidean distance is used in the De-Mix strategy to approximately preserve the kernel distance). The experiments have been performed ten times and the average accuracies are reported. In the figure, the X axis value means the total number of training samples, including the original training set and the newly added samples from the test set. From Figure 6, we can see the Adaptive Selective Learning yields the best accuracy in all regions. We analyze the results in detail as follows.

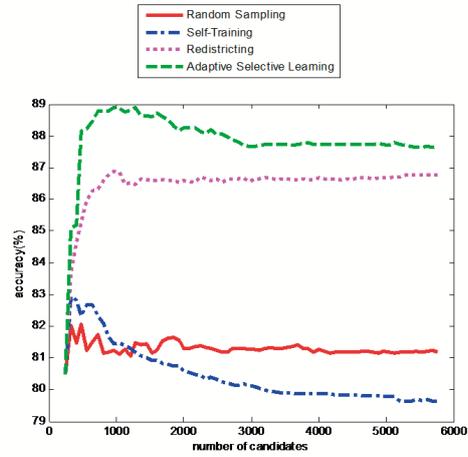
First, the Random Sampling may performance better than Redistricting at the beginning, for example, in the central region, but not as good as Adaptive Selective Learning. This is because the Redistricting only pays attention to the class distribution but not the instance selection, thus it sometimes may be worse than Random Sampling. The Adaptive Selective Learning not only considers the class distribution but also the instance selection. Thus, it is always performs the best compared to the other three algorithms.

Second, Self-Training only selects the instance with high confidence. It is clearly that the higher the confidence of an instance, the further of its distance to the decision boundary. Thus, Self-Training only selects those instances which are far away from the decision boundary. These kinds of instances cannot continuously contribute to the classifier induction while adding more redundant instances. Therefore, the accuracy of Self-Training will decrease when more instances are added. As discussed in details in Section 4, the Adaptive Selective Learning does not only select those with higher confidence but also maintain a good class ratio between two different class labels craters and non-craters. Hence it performs better than Self-Training.

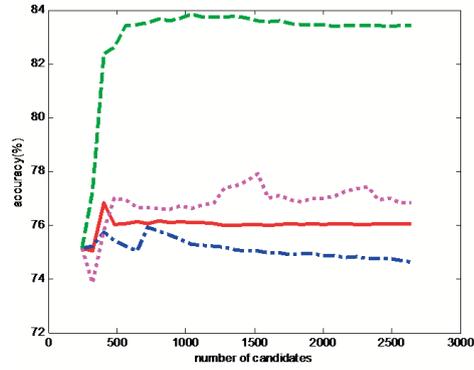
Third, the difference between Redistricting and Adaptive Active Learning illustrated in the central region (Figure 6B) is larger than those in the west region and the east region. This shows that the quality of candidates in the central region is not as good as those in the west region and the east region. The reason of this is due to much diversified surface morphology of the central region, compared to other two regions.

Finally, all algorithms cannot keep improving detection accuracy after too many new test instances are added into the training set. The prediction accuracy of the derived classifiers from the training set will be gradually decreased due to the errors introduced during automatic labeling process. This phenomenon is clearly demonstrated in Figure 6C. The accuracy values are decreased after more than 1,500 candidates are added into the training set.

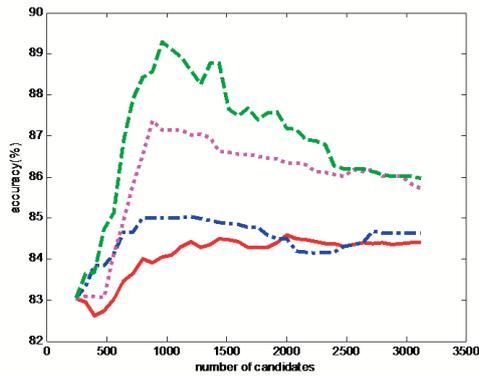
We further compare the four algorithms with the state-of-the-art semi-supervised learning algorithm S3VMs [Chapelle, et al., 2006]. Figure 7 summarizes the performance of five algorithms using LIBSVM, Random Sampling, Self-Training, Redistricting, S3VM, and Adaptive Selective Learning. We compare these algorithms by accuracy, precision, and recall. Column one is the number of can-



(A) West Region



(B) Central Region



(C) East Region

Figure 6: Performance results of the Random Sampling, Self-Training, Redistricting, and adaptive Selective Learning; 1st row: the west region, 2st row: the central region, 3rd row: the east region.

	# of Candidates	Evaluation	LIBSVM	Random Sampling	Self-Training	Redistricting	S3VMs	Adaptive Selective Learning
West	6428	Accuracy	80.48%	82.05%	82.92%	86.68%	84.78%	88.91%
Region		Precision	81.54%	83.34%	84.79%	87.01%	86.79%	90.46%
		Recall	77.31%	78.99%	80.86%	83.83%	83.33%	85.94%
Central	2813	Accuracy	75.15%	76.82%	75.95%	77.93%	80.21%	83.86%
Region		Precision	75.56%	78.13%	80.02%	82.72%	83.26%	89.86%
		Recall	71.21%	74.81%	73.75%	75.55%	74.85%	80.83%
East	3301	Accuracy	83.03%	84.91%	85.03%	87.37%	84.21%	89.28%
Region		Precision	79.56%	80.83%	81.12%	82.45%	80.08%	83.36%
		Recall	84.45%	85.54%	86.61%	89.9%	87.56%	91.34%

Figure 7: Crater detection accuracy results of the LIBSVM only, Random Sampling, Self-Training, Redistricting, S3VMs, and Adaptive Selective Learning.

didates showing how many candidates detected in each region. Each column is the best result that the algorithm can achieve. LIBSVM serves as the classifier used in the other three algorithms except for S3VMs. From the table, we can see, compared to only using LIBSVM, Random Sampling can mostly improve 1.57%, 1.67%, 1.88% in accuracy in three regions, respectively; Self-Training can mostly improve 2.44%, 0.80%, 2.00% in accuracy; Redistricting can mostly improve 6.20%, 2.78%, 4.34% in accuracy; S3VMs can mostly improve 4.30%, 5.06%, 1.18% in accuracy; and Adaptive Selective Learning can mostly improve 8.43%, 8.71%, 6.25% in accuracy. The Adaptive Selective Learning again has improved the most in precision and recall. All in all, our approach, Adaptive Selective Learning, performs the best. Furthermore, we observe that the improvement in the central region is greater than the other two regions using our approach because Adaptive Selective Learning can intelligently generate different training sets according to different test sets.

5.3 CONCLUSIONS AND FUTURE WORK

This paper aims at improving the detection rate for auto-detection of small craters in high-resolution images of planetary surfaces. The proposed Adaptive Selective Learning uses innovative methods on training set construction, using active learning and semi-supervised learning. Significant performance gain has been observed in our case study site on Mars.

In future work, we should make analysis on the initial training set. Analyze the instances in the initial training set when transfer the knowledge from the test set and discard the instances which are not compatible with the test set. In addition, how to intelligently decide the thresholds for Majority Vote Strategy, De-Mixed Strategy should be further studied.

5.4 ACKNOWLEDGMENTS

The work is partially supported by the NASA grant NNX09AK86G. The authors would like to thank Ph.D. students Yang Mu and William Isaac Miller of the Knowledge Discovery Lab, Computer Science Department, University of Massachusetts Boston for informative discussions on the algorithm design.

References

- R. Martins, P. Pina, J. S. Marques, and M. Silveira. Crater detection by a boosting approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1):127-131, 2008.
- C. Papageorgiou, M. Oren, and T. Poggio. (). A general framework for object detection. In *Sixth International Conference on Computer Vision*, pages 555-562, 1998.
- S. Dasgupta and D. Hsu. Hierarchical sampling for active learning. In *ICML*, pages 208-215, 2008.
- S. Ertekin, J. H. 0002, L. Bottou, and C. L. Giles. Learning on the border: active learning in imbalanced data classification. In *CIKM*, pages 127-136, 2007.
- B. Settles. Active learning literature survey. *Computer Sciences Technical Report 1648*, University of Wisconsin-Madison, 2009.
- D. Yarowsky. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 189-196, 1995.
- M. Culp and G. Michailidis. An iterative algorithm for extending learners to a semisupervised setting. In *The 2007 Joint Statistical Meetings (JSM)*, 2007.
- G.R. Haffari and A. Sarkar. Analysis of semi-supervised learning with the Yarowsky algorithm. In *23rd Conference on Uncertainty in Artificial Intelligence (UAI)*, 2007.
- Kim, J. R., Muller, J.-P., van Gasselt, S., Morley, J. G., Neukum, G., 2005. Automated crater detection, a new tool for mars cartography and chronology. *Photogramm. Eng. Remote Sens.* 71 (10), 1205-1217.
- Viola, P., Jones, M., 2004. Robust real-time face detection. *Inter. Journal of Computer Vision* 57 (2), 137-154.
- D. Angluin. Queries and concept learning. *Machine Learning*, 2:319-342, 1988.
- D. Cohn, L. Atlas, R. Ladner, M. El-Sharkawi, R. Marks II, M. Aggoune, and D. Park. Training connectionist networks with queries and selective sampling. In *Advances in Neural Information Processing Systems (NIPS)*. Morgan Kaufmann, 1990.
- D. Cohn, L. Atlas, and R. Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201-221, 1994.
- D. Lewis and W. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3-12. ACM/Springer, 1994.
- R. Lomasky, C.E. Brodley, M. Aernecke, D. Walt, and M. Friedl. Active Class Selection. *Machine Learning: ECML 2007*, 4701: 640-647, 2007.
- Leroy, B., G. Medioni, A. E. Johnson, and L. H. Matthies, "Crater detection for autonomous landing on asteroids," *Image Vis. Comput.*, vol. 19, no. 11, pp. 787-792, Sep. 2001.
- Honda, R., 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*. New York: Springer-Verlag, 2002.
- Cheng, Y., A. E. Johnson, L. H. Matthies, and C. F. Olsen, Optical landmark detection for spacecraft navigation, in *Proc of the 13th AAS/AIAA Space Flight Mechanics Meeting*, Ponce, Puerto Rico, pp. 1785- 1803, AAS 03-224, 2003.
- Barata, T., E. Ivo Alves, J. Saraiva, and P. Pina, Automatic recognition of impact craters on the surface of Mars, in *Proc. ICIAR*, Porto, Portugal, pp. 489-496, 2004.
- Hough, P. V. C., Method and means for recognizing complex patterns, U.S. Pat. Off., Washington, DC, U.S. Patent 3069654, Dec. 18, 1962.
- Earl, J., A. Chicarro, C. Koeberl, P. Giorgio Marchetti, and M. Milnes, "Automatic recognition of crater-like structures in terrestrial and planetary images," presented at the Workshop

Role Volatiles and Atmospheres Martian Impact Craters, Lunar Planetary Inst., Houston, TX, #3053, 2005.

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

Vinogradova, T., M. Burl, and E. Mjosness, Training of a crater detection algorithm for Mars crater imagery, in Proc. IEEE Aerosp. Conf., vol. 7, pp. 7-3201-7-3211, 2002.

Plesko, C., S. Brumby, E. Asphaug, D. Chamberlain, and T. Engel, Automatic crater counts on Mars, Lunar and Planetary Science XXXV, Lunar Planetary Inst., Houston, TX, #1935,(CD-ROM), 2004.

Wetzler, P. G., B. Enke, W. J. Merline, C. R. Chapman, and M. C. Burl, Learning to detect small impact craters, in Proc. 7th IEEE WACV/MOTION, vol. 1, pp. 178-184, 2005.

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

W. Ding, T. Stepinski, L. Bandeira, R. Vilalta, Y. Wu, Z. Lu, T. Cao, "Automatic Detection of Craters in Planetary Images: An Embedded Framework Using Feature Selection and Boosting," the 19th ACM International Conference on Information and Knowledge Management (CIKM 2010), Toronto, Canada, October, 2010. Chih-Chung Chang and Chih-Jen Lin, LIBSVM : a library for support vector machines, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>

HRSC Data Browser. <http://europlanet.dlr.de/node/index.php?id=209>, 2010.

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

K. L. Tanaka. The stratigraphy of Mars. *Journal of Geophysical Research*, 91:E139-E158, November 1986.

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

Kohavi, Ron . "A study of cross-validation and bootstrap for accuracy estimation and model selection". *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence* 2 (12): 1137-1143, 1995.

R. Martins, P. Pina, J. S. Marques, and M. Silveira. Crater detection by a boosting approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1):127-131, 2008.

C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision*, pages 555-562, Bombay, India, 1998.

Olivier Chapelle, Vikas Sindhwani, and S. Sathiya Keerthi. Branch and bound for semisupervised support vector machines. In *Advances in Neural Information Processing Systems (NIPS)*, 2006.