Lunar Crater Detection Using Bag of Features

David Leblanc
Dr. George Bebis
05/30/2010

1. Introduction:

This is a summary of the approach used to detect and classify images of craters from the moon. The approach is based on the Csurka and Dance paper[2]. The method was implemented using SURF features using a library called OpenSURF[1], which is an extension of the OpenCV Library. This report includes an explanation of the method used, the experiments done, the results and a discussion.

2. Method:

So far, the method used follows the paper from Csurka and Dance[2]. This paper uses K-means clustering to create a descriptive visual dictionary. This visual dictionary is used to describe images in a way which they can be classified. Two methods were used to cluster the data in this experiment: first using K-means clustering, and second, using hierarchical clustering. The steps of this method, using K-means clustering, are as follows:

1. Obtain images of craters and of non-craters, and split them into three sets: a training set (60%), a validation step (20%), and a test set (20%).
2. Extract features (in this case, SURF features) from a set of images of craters, and non-craters.
3. Cluster the features and use the cluster centers as “visual words”. This creates of a dictionary of visual words of size *K* (the number of clusters).
4. Build histograms that represent each crater and non-crater images in terms of those visual words, and normalize them.
5. Once all the training images have been expresses as histograms, attach a label to each image or histogram. In this case, '1's for craters and '-1's for non-craters.
6. Input the training data into the SVM classifier and train it to find the distinction between crater and non-crater histograms.
7. Using the trained classifier and the validation set of images (expressed in histograms), attempt to predict the validity of the system by calculating the precision of correct classification of both validation craters and non-craters. Save the results and repeat the steps a number of times *n* from step 3 and keep the dictionary that gives the best validation results.
8. Given that the best dictionary was found in steps 3 – 7, the test images can then be expressed as histograms, and predicted using the SVM classifier.

In this case, steps 1 – 7 are done offline. The validation set is used to find the optimal performance of the K-means clustering algorithm. The reason we need to repeat the clustering many times is that K-means produces random clusters every time. The results of the test data should in theory follow the validation results.

A) Classification using K-means Clustering:

The images used for this experiments have been manually extracted from images of the Moon's surface. The images gathered do not all have the same dimensions so images that are very small are enlarged to ensure that more features can be found. SURF features are extracted from each image and are saved into a matrix. The matrix contains both the SURF vectors of the crater and non-crater data. The next step is to construct a visual dictionary of features. To accomplish this task, the matrix containing the SURF data is clustered based on the number of clusters *K*. Because the K-means clustering algorithm chooses it's initial conditions randomly for clustering, some of the randomly generated points may fall outside of the data space, or in between two clusters. This means that a number of cluster points may not have any data associated with it. An important step in the algorithm is to find and remove any clusters that are empty.

Once all the empty clusters have been removed, the dictionary of visual words is constructed by finding the center points of all the non-empty clusters. The cluster centers are simply the average of all the points in a given cluster. All the points inside the clusters are summed up, and divided by the number of points found in that cluster. With all the cluster centers found, the next step is to build the histograms of all the training images.

Each cluster center point is a representative bin in the histograms. To build the histograms, for each image, each SURF feature is matched to a “word” from the visual dictionary by computing the shortest euclidean distances between that feature and the cluster centers. The number of features in an image found for each bin is counted, and the histogram is represented as a vector of size *k* which is the number of visual words. This histogram vector is then normalized so that it's magnitude is 1. Each image from the training set is represented as a normalized histogram vector.

The histograms are then attached numeric labels (1 for craters and -1 for non-craters), and are fed to the support vector machine or SVM for training. This classifier will try to find the separation between the crater class, and the non-crater class. In the same manner, the histograms for the validation data are constructed, and given to the SVM classifier for prediction. The prediction will return a numeric value of either 1 or -1, which should match the label previously fed to the SVM for each class. The number of positive matches are counted and the precision for the validation is computed. These results need to be saved, and the entire process should be repeated. A new dictionary is constructed using the same *K* values for the clustering, and the histograms are rebuilt based on that new dictionary. The SVM is retrained, and the validation step repeated. The results of the validation need to be compared to the previous iterations. If they are better, then the dictionary is kept, otherwise it is discarded.

The number of iterations performed affects the results quite a bit. Because the dictionaries are essentially built randomly because of the K-means clustering algorithm, more iterations yield a better chance of creating a more discriminating dictionary that will perform better overall. The best dictionary is kept and will be the one used for testing. In the same way, test images are converted to histograms, and classified using the best trained SVM.

B) Classification using Hierarchical Clustering:

The general main steps when using hierarchical clustering are essentially identical to K-means. The hierarchical clustering method builds a tree of clusters, where each node represents a cluster. A good aspect of this clustering algorithm is that the tree is built in a very systematic way, therefore, for a given data set, there is a unique tree solution for it. This means that the validation step then becomes irrelevant when using this method. This is good alternative because the validation data can then be added to the training data, which will result in a more extensive training.

Determining the number of clusters is done by cutting the tree. In this tree, symbolically, the leafs are each individual data points and the root is one large cluster which includes all data points. The nodes in between are clusters of varying sizes. It is then possible to obtain an exact *K* number of clusters by cutting the tree in a specific manner. For any value *K*, there is only one way to cut the tree to get that exact number of clusters. This makes the clustering and dictionary building steps much simpler. The system then becomes more or less constant, and repeatable. With this established, there is no need to check for empty clusters in the clustering step. The histograms are built using the method described previously, and all other steps are implemented the same way.

3. Experiments:

The set of images used includes manually extracted patch from images of the Moon's surface. The set includes 169 images of craters and 200 images of non-craters. This is a relatively small set of images for the task of classification, but for the purpose of this experiment, it is sufficient. Gathering more image patches could easily be accomplished if needed. Figure 1 shows a sample of crater images, and Figure 2 shows a sample of images of non-craters used for training, validation, and testing.

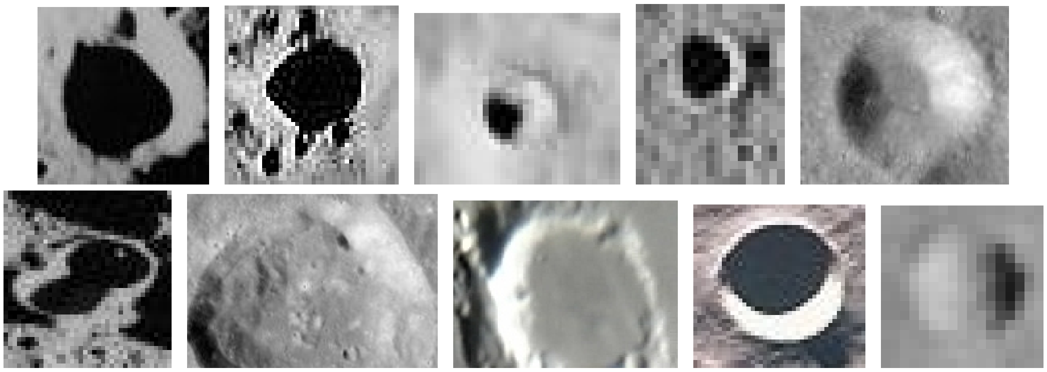


Figure 1. Sample of images of craters

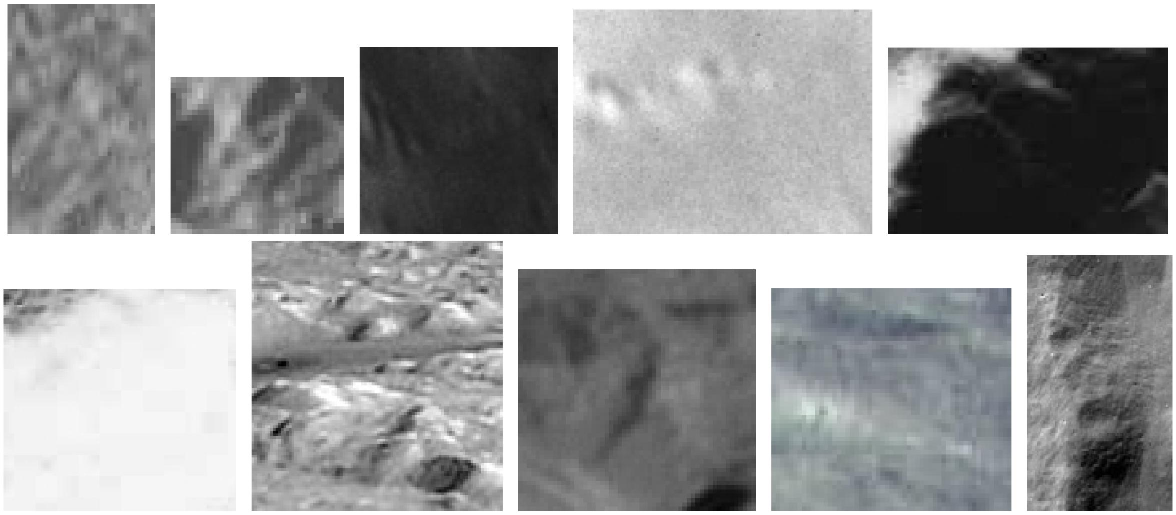


Figure 2. Sample of images of non-craters

The sample provides a wide range of illumination changes, perspectives, and scales for both craters and non-craters. This does present a challenge given the small size of the set. Some images are more or less blurry, and generally, images range from very small in size (10x10 pixels) to larger (200x200 pixels). For the experiments, the smaller images are scaled up to a more reasonable size to ensure that features can be extracted, and to provide a sense of normalization.

The first step of the method is to extract features from the training images. Figure 3 shows images with SURF features extracted, using the OpenSURF library. These features are similar to the popular SIFT features, only they are extracted faster since they are vectors of length 64. Once all the features were extracted, clustering the data, and building the visual dictionaries.

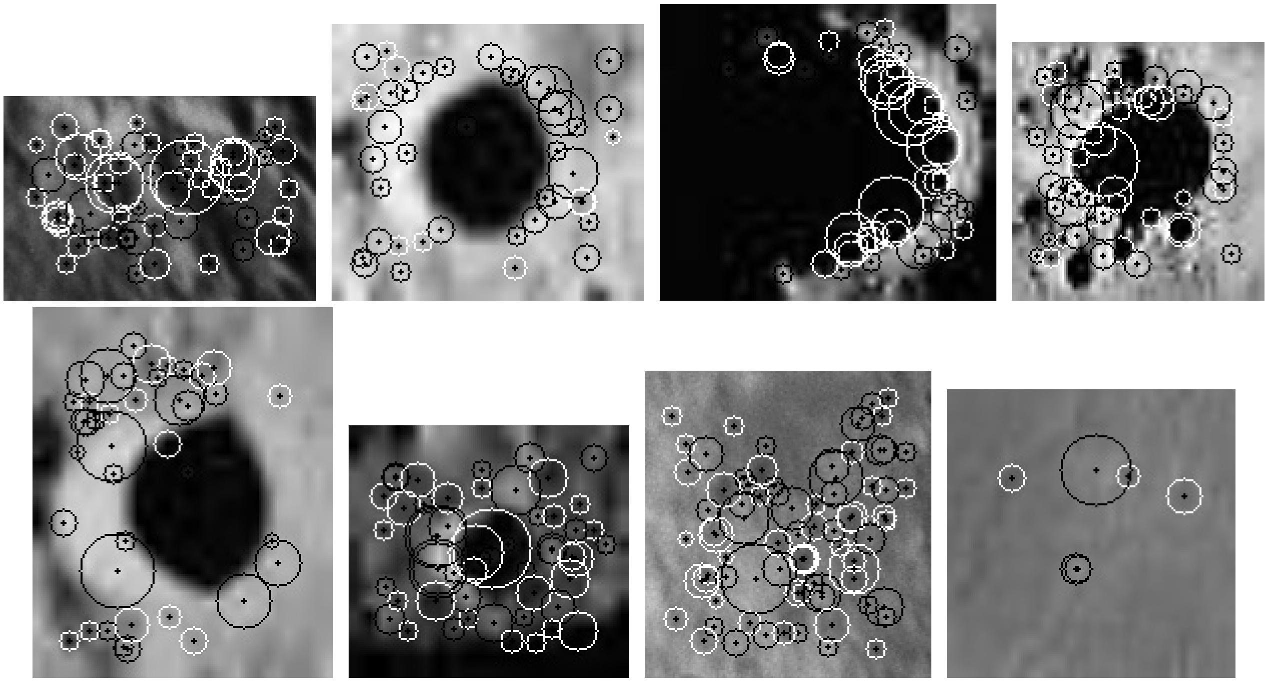


Figure 3. Images of the moon with SURF features extracted

Two main experiments were conducted: one using K-means, and the other using hierarchical clustering. For both experiments, five trials were generated by randomly shuffling the data images. For K-means clustering, the data is split as follows: 60% for training, 20% for validation, and 20% for testing purposes. For the hierarchical clustering, the validation is omitted for reasons stated above, so the data was split as 80% for training, and 20% for testing. Because the two experiments do not split the data the same way, the results cannot be directly and fairly compared. But some comparisons can be made because, arguably, the 20% used for validation in K-means is part of the training because it helps ensure a better clustering of the data.

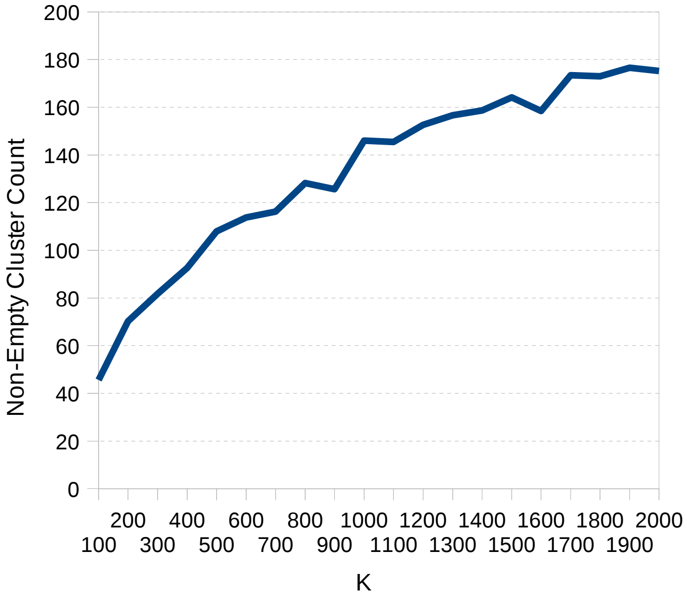
The number of features extracted for training when using K-means (60%) was approximately 13,000 to 14,000 (depending on the trials), as opposed to the hierarchical clustering experiment (80%) which extracted on average 17,000 features from the training data. For the K-means experiment, each trial was ran with 20 different *K* values ranging from 100 to 2,000 (in steps of 100). For each *K* values, the validation step was performed eight times, to find the best dictionary possible. This number was picked somewhat arbitrarily, but seem to offer a good trade-off of computing time and accuracy. Increasing this number may increase the accuracy of the system, but for purposes of this application, eight times seemed to be an appropriate compromise. Obviously, increasing this number increases the complexity, memory requirements, and execution time, but since the training is done offline, this would not be much of an issue.

For the experiment using hierarchical clustering, each trial was performed with 100 cluster values *K*, increasing in steps of 20 (from 20 to 2000). Running these experiments can be done much faster because the tree only needs to be built once. Building the tree takes a few minutes, and once built, it can be cut in different ways, which only takes a few seconds.

This experiment was done using the “C Clustering Library”[3]. This library provides four different hierarchical clustering techniques: pairwise single-linkage, complete-linkage, average-linkage, or centroid-linkage. This experiment was done using single-linkage because it requires less computation and is more memory efficient. The memory requirements of the other algorithms are too demanding for the amount of data for this application. The computer used to test this method had only 1GB of memory, which is insufficient for the requirements of other linkage techniques.

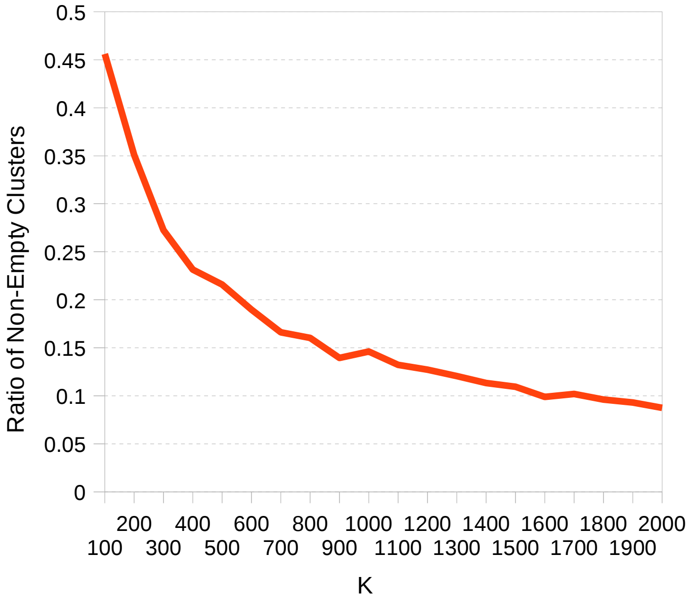
4. Results:

The results for the K-means clustering appear to vary quite a bit. One of the most important aspects to consider in the case of this method is the number of non-empty clusters. This represents the number of visual keypoints in the dictionaries. This parameter is hard to control because the K-means produces random results. Because the data set is relatively smaller, the parameter *K* produces a large number of empty clusters. Graph 1 shows the results of the number of non-empty clusters based on the parameter *K*.



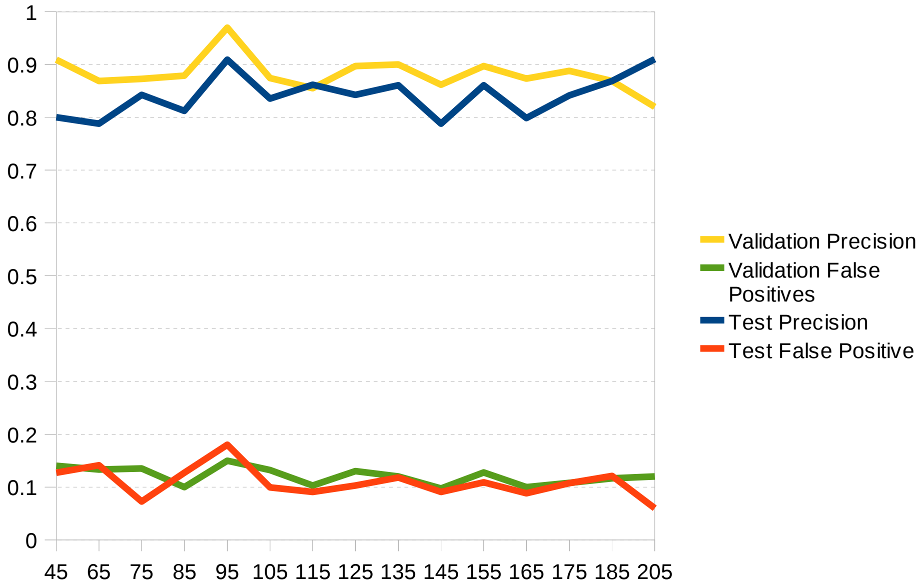
Graph 1. Non-empty Clusters as a function of K

Another way to see this problem is the ratio of non-empty clusters to *K* as *K* increases. Graph 2 clearly shows that increasing *K* does not result in a significant increase in non-empty clusters. These are the average results for all five experiments.



Graph 2. Ratio of non-empty clusters to values of *K*

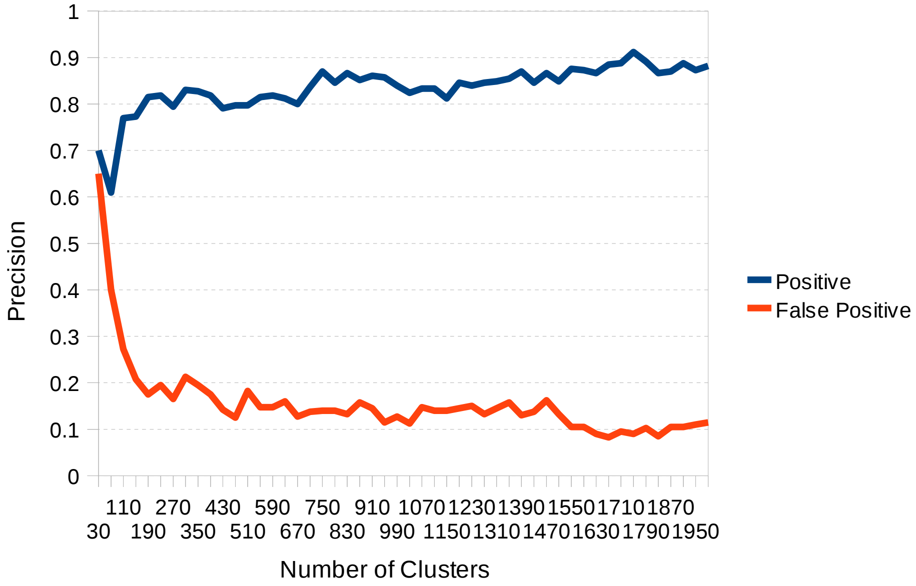
These results are to be expected because the data set is relatively small and the K-means is done randomly. There is some variation in these results because the validation step is performed many times for each value of *K*. The validation step will keep the dictionary that performs the best on the validation data. The relationship between the number of non-empty clusters and the validation results is impossible to predict. For example, two cases may have the same number of non-empty cluster, but the way they were clustered yields very different results. This can be shown in the graph below. Graph 3 shows the precision for the validation and test data. The graph also shows the false positive rates as a function of the number of non-empty clusters.



Graph 3. Results using K-means clustering

Graph 3 shows how much K-means classification varies as the size of the dictionary changes. As stated previously, this variation is due to the fact that K-means chooses it's cluster centers randomly. This results in very different visual dictionaries from one validation step to the other. The precision results for the test vary from 80% to 90% for the true positives. The yellow line represents the results obtained from the validation step (keeping the best results), so it is to be expected that the validation precision is slightly higher. From this data, it is really impossible to determine any type of trend in the results.

The results for hierarchical clustering are much more stable. Graph 4 shows these results. It shows both the true positives and false positives rates, and is a function based on the number of clusters used. The number of clusters is directly related to the size of the dictionary. In the case of hierarchical clustering, there are no empty clusters because the tree is built before the clustering is done. The tree is simply cut in different ways, as explained previously.



Graph 4. Results for hierarchical clustering

As shown in Graph 4, the trend appears to show that as the number of clusters increase, the true positive rate increases steadily, and the false positive rates decrease slowly. The results for this method are a lot less noisy, and more reliable. Based on this graph, the results appear to be the best at a number of clusters set to or around 1700. Figure 4 shows images of craters that were correctly classified as craters (true positives), and Figure 5 shows crater images that were incorrectly classified at a *K* value of 1700.

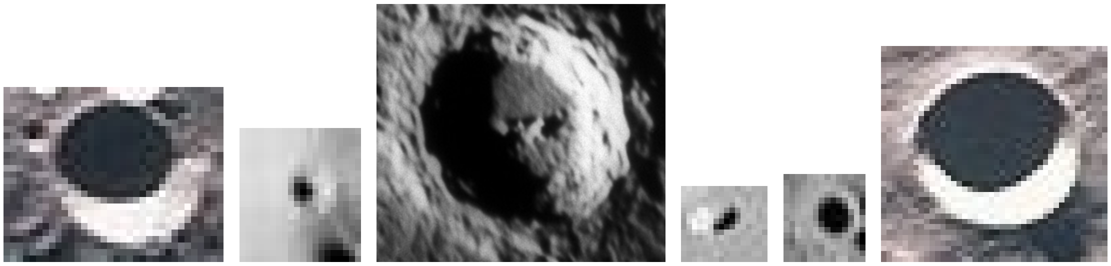


Figure 4. Examples of crater images that were correctly classified (true positives) at *K* = 1700

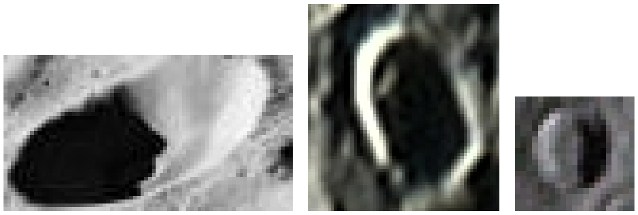


Figure 5. Crater images than were incorrectly classified at *K* = 1700

From the images above, we can see that the method appears to be robust to large scale changes in the crater size, and also robust to illumination changes up to a certain point. Figure 4 appears to show that the method seems to no work as well on images of crater with different perspective angles, or images that are not very sharp, or have very poor lighting conditions. Overall, the method results in an accuracy of over 0.90 at a *K* value of 1700. Figure 6 and Figure 7 are non-crater images that were correctly and incorrectly classified at *K* = 1700.

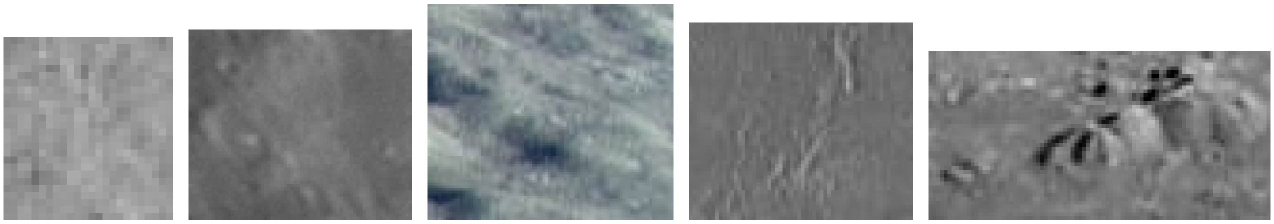


Figure 6. Examples of non-crater images that were correctly classified at *K* = 1700.

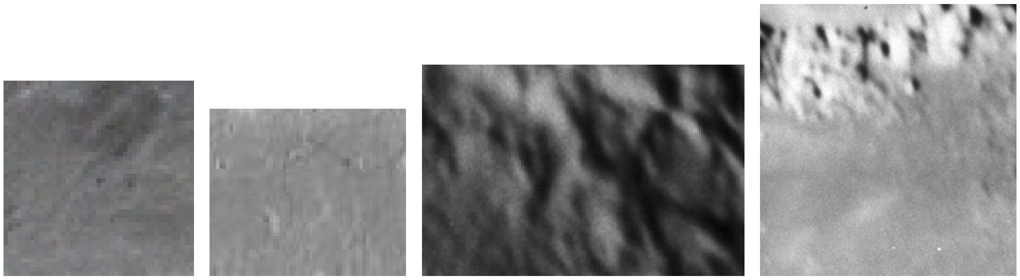


Figure 7. Non-crater images that were classified as craters (false positives) at *K* = 1700.

From figure 6 and 7, it's harder to uncover the reasons why certain images were wrongfully classified as craters. Arguably, some crater-like features can be found in some of the images in Figure 7, but most likely there are other reasons why these images were classified as craters. The false positive rate for this systems at *K* = 1700 is about 0.09.

5. Discussion

The results above appears to show that hierarchical clustering would be a better method for classification of craters, using a small set of training images. The K-means is simply too noisy, random, and unreliable to predict which *K* values will yield the best results. Increasing the values of *K* does not necessarily result in better classification. With K-means, the results depend solely on how the initial cluster centers are chosen. Because the features have high dimensionality, and the set of training images is very small for this experiment, obtaining a good clustering representation may require many iterations to find optimal results.

Another issues is that since the set of training, validation and testing images are so small, there is no guarantee that a given dictionary produced through K-means clustering will perform well on both validation and test sets. Graph 3 shows that on average, the validation and test sets tend to follow each other, but in individual cases, this may not be true. A certain dictionary may perform poorly on the validation set, but results in the best classification for the test set. Because it did not perform well on validation, then it is rejected. The main factor contributing to this is the very limited sets of images. Increasing the number of images would certainly yield better results.

Performing the validation step, although done offline, is very inefficient. In the experiments performed above, for each value of *K*, eight validation iterations were performed to increase accuracy. Inefficiency in run-time for the offline training is not much of an issue, but it is interesting to note that performing these tests requires much more computation time than using hierarchical clustering.

Training the system using hierarchical clustering is more efficient since the steps are to build the tree and cut it in a way to get a precise number of clusters. This method is more reliable because for a given set of data, the tree will always be built and cut the same way. Another important aspect is that hierarchical clustering does not produce any “empty” clusters. This allows the visual dictionary to be much larger and more descriptive. Increasing the number of words, as shown in Graph 4, appears to increase the classification accuracy. This comes at a cost; increasing the size of the dictionary will increase the computation time for building the histograms. But generally speaking, hierarchical clustering will yield much better results on average.

In an attempt to improve the accuracy of the system, another test was implemented, by using a number of different dictionaries in the hierarchical tree. Testing images on different sized visual dictionaries may result in better accuracy. But in the case of this experiment, this did not improve the accuracy by much or at all, because the training set is so small. The computational cost of using multiple dictionaries will obviously be more expensive, so in this case, this may not be the best solution.

6. Conclusion

Overall, the results are very good considering very small set of training images. Normally, applications that use this method generally use a much larger set of images. Increasing the number of images would definitely improve the accuracy of lunar crater detection. More training images will result in more extracted features, which will make the clustering step more descriptive and accurate. The hierarchical clustering method is definitely a better choice than the K-means clustering for this application. Increasing the training set would help the K-means clustering, but hierarchical will still perform better and give more reliable results.

Other ways to improve the accuracy of the system would be to use SIFT features instead of SURF features. SURF features are extracted faster, but are not as robust on extreme lighting conditions and affine variations. The SIFT features would fix these problems since they are more robust to these types of transformations. Although they are not as fast as SURF, the slight increase in computation should not affect the system speed much. The most computationally expensive part of the method is the training, which is done offline, so this is not much of a concern.

The hierarchical clustering method currently used is pairwise single-linkage. Trying different clustering methods may yield better results. Other linkage methods are more memory and computationally extensive, but again, this is done offline. A better clustering will result in a more descriptive visual dictionary, which in return will result in a better detection accuracy.

References

[1] Evans, Chris. "The OpenSURF Computer Vision Library." *Chris Evans Development*. N.p., 2010. Web. 05 Jan 2010. <http://www.chrisevansdev.com/computer-vision-opensurf.html>.

[2] Csurka, Gabriella, and Christopher Dance. "Visual Categorization with Bags of Keypoints." (2004): Print.

[3] de Hoon, Michiel, Seiya Imoto, and Satoru Miyano. "The C Clustering Library." *Open Source Clustering Software*. N.p., 05 Apr 2010. Web. 31 May 2010. <http://bonsai.ims.u-tokyo.ac.jp/~mdehoon/software/cluster/software.htm>.