

# A Trainable Tool for Finding Small Volcanoes in SAR Imagery of Venus

M.C. Burl<sup>†‡</sup>, U.M. Fayyad<sup>‡</sup>, P. Perona<sup>‡§</sup>, P. Smyth<sup>‡</sup>, and M.P. Burl<sup>‡</sup>

<sup>†</sup> California Institute of Technology 116-81, Pasadena CA 91125, U.S.A  
e-mail: burl,perona@systems.caltech.edu

<sup>‡</sup> Jet Propulsion Laboratory

<sup>§</sup> University di Padova, Italy

October 25, 1993

## Abstract

Volcanoes appearing in SAR images of the surface of Venus are reliably detected and located automatically by a system designed to handle general visual pattern recognition problems. The system is based on classical techniques of linear filtering and supervised statistical pattern classification and is trained for the specific task using a small number of examples provided by experts; a set of matched filters is synthesized from the training examples using principal components analysis and normalization techniques. System-level issues such as (a) collecting reliable training examples, (b) training and performance evaluation in the absence of a ground truth, (c) dimensioning the feature space, are addressed in detail. Tests conducted on 4 images containing 163 volcanoes show detection vs. false alarms performance that is comparable to that of trained human observers.

Keywords: pattern recognition, vision, principal components, volcanoes, Venus, SAR, detection, machine learning, classification

# 1 Introduction

## 1.1 Automatic analysis of planetary images

The volume of image data collected by spacecraft has reached a level that precludes the traditional approach of ‘manually’ examining each collected image. ‘The end-users - the scientists - can no longer perform global or comprehensive analyses effectively, making automated tools that locate features of interest in large image databases a necessity.

Our goal is to develop a general system for aiding scientists in locating patterns of interest in image data. We envision such a system working in the following way. The scientist will specify what to look for simply by using a mouse to point and click on instances of the target object in a number (the fewer, the better) of sample images. The tool will automatically ‘learn’ relevant characteristics of the target object by analyzing the examples provided by the scientist, and will then search through a large database to find all instances of the target object.

Such a tool would of course be applicable to many domains extending beyond remotely - sensed image analysis - for example to applications such as medical image analysis, security (face recognition), industrial inspection, and defense. The approach we propose is based on combining pattern recognition and machine learning techniques to generate automatically from training data a classifier that can locate and identify objects of interest in imagery.

In this paper we describe the prototype system that we have developed and we describe its application to the problem of automatically locating small volcanoes in the synthetic aperture radar (SAR) imagery of Venus collected by NASA and JPL’s Magellan spacecraft [6]. This work was conducted in collaboration with scientists at Brown University’s Department of Geological Sciences.

Locating small volcanoes in the Magellan SAR data is particularly suitable as a testbed domain for several reasons:

- Over 30,000 images containing an estimated  $10^6$  small volcanoes ( $< 15$ km in diameter) were returned by Magellan’s first mapping cycle. In fact, the Magellan mission has resulted in a data set which is larger than that gathered by all previous planetary missions combined. Planetary scientists are literally swamped with data, and it is clear that manual examination and characterization of this many objects is prohibitive.
- Understanding clustering characteristics and the global distribution of the volcanoes is fundamental to understanding the regional and global geologic evolution of the planet [1]. This study can potentially provide the data necessary to answer basic questions concerning the geophysics of Venus, which is of particular interest since geologically, Venus is Earth’s sister planet.
- The Magellan application is the type of problem that is increasingly encountered by data analysts in remote-sensing, as well as in other fields, as databases grow larger. Quick access to specific features of interest that are difficult to define directly is essential.
- Geographical ‘objects’ present no well-defined visual features that are easily transformed into geometrical tokens. Much of the work in visual pattern recognition and object recognition in computer vision so far has focused on man-made objects where

well defined regular structures such as bolts, lines, and holes may be measured reliably using general-purpose edge and line detectors.

- The Magellan data set is the first truly digital planetary data set; anyone with a desktop computer and a CD-ROM can easily access the entire set of data products.

## **1.2 Related Work on the SKICAT Program**

To emphasize that current supervised machine learning algorithms can yield significant results on real-world problems involving image data, we briefly review prior work on the Sky Image Cataloging and Analysis Tool (SKICAT) problem [5, 4, 10], which involves finding and classifying sky objects (stars, galaxies, etc, ) in digitized images of the Second Palomar Observatory Sky Survey (POSS-II).

The POSS-II survey consists of approximately 3,000 digitized photographic plates, each consisting of 23,040 x 23,040 16-bit pixels. The survey contains on the order of  $10^8$  sky objects. Using public-domain image processing and region growing algorithms, attributes defined by astronomers are measured for each object in a set of training images. The attributes include intensity, area, ellipticity, statistical moments, and so forth. Training data are constructed by asking astronomers to manually classify each of objects in the training set. Decision-tree learning algorithms [5, 3, 9] are then applied to learn a mapping from the measured attributes to the desired classification. For objects that are too faint for astronomers to classify, the training data is obtained from higher resolution images or previous small-scale surveys covering the corresponding portion of the sky. This allows the learning algorithm to learn to classify objects that are too faint for the astronomers themselves to classify by visual inspection.

The Sky Image Cataloging and Analysis Tool (SKICAT) was developed in collaboration with personnel from Caltech's Astronomy Department. We have been able to achieve a classification accuracy rate of about 9470, well above the 90% rate required for reliable scientific analysis. In addition to the tremendous savings in the required manpower, we can catalog objects that are at least one magnitude fainter than any objects cataloged in any comparable sky survey to date. Since the majority of objects in an image are too faint to be classified manually or by traditional hand-coded algorithms, the SKICAT-generated catalog will contain three times as many objects as would have been possible by manual or traditional computational cataloging techniques. This is an excellent example of a machine learning approach that enabled scientists to automate a laborious procedure as well as extract more data at no extra cost. Furthermore, the SKICAT classifier represents an objective, examinable, repeatable approach to sky object classification, providing a uniform classification throughout the entire catalog over the 108 sky objects.

## **1.3 Contents of the paper**

An important factor in the success of SKICAT and a fundamental difference between SKICAT and the Magellan problem is the existence in the astronomy field of a set of well-defined, invariant features, which astronomers have worked on for decades. For Magellan and many other visual pattern recognition problems, no such "ready-made" feature set is available.

Therefore, to develop a general system for locating patterns of interest in image data, we must address the problem of finding good mappings from pixel space to features (feature synthesis, or feature 'learning'), as well as the problem of finding a good mapping from feature space to class labels (classifier learning). By following a well-principled, domain-independent approach to the feature synthesis problem, we hope to produce an algorithm that is effective, easy to use, and portable with minimal effort to other planetary database exploration tasks and ultimately to generic visual recognition problems of widespread interest, e.g., product inspection (finding defects on parts coming out of a production line) and security (finding human faces in cluttered scenes).

Because our goal is to develop a general system for aiding scientists in locating patterns of interest in image data, the prototype system we have developed relies extensively on learning from examples provided by the scientists. In some domains (including MagellanSAR), the noise level, image resolution, and other factors make it impossible for the scientist to label all objects of interest with 100% confidence. This presents two orders of problems: (a) the training set is not completely reliable, and (b) there is no "ground truth" to evaluate the performance of the algorithm. These issues are discussed in Section 3.

In Section 4 we define the overall structure of our prototype system. The system uses a focus of attention (FOA) mechanism to identify quickly candidate regions that may contain the pattern of interest. The regions of interest (ROIs) selected by the focus of attention are compared to the scientists' examples to separate positive-examples from counter-examples. Next, in the feature synthesis, or feature learning, stage (FL), the system uses principal component analysis to learn relevant features from the examples. The learned features are used to map the examples and counter-examples from pixel space to a lower-dimensional feature space. Supervised classification algorithms are applied to learn the appropriate mappings from feature space to class label (volcano vs not-volcano). The focus of attention, feature synthesis, and classification algorithms are described in detail in Section 5.

The prototype system has been applied to a limited set of Magellan images. The performance of the algorithm is quite promising and is, in fact, close to the performance level of expert geologists. The performance evaluation is presented in Section 6.

Finally, in Section 7 we summarize the current status of our effort to develop a trainable tool for locating patterns of interest in image data. We also discuss some of the open issues that we intend to address in future research.

## 2 Magellan Imagery

A fundamental objective of the Magellan mission was to provide global mapping of the surface of Venus. The mapping was performed using synthetic aperture radar (SAR) because of its ability to penetrate the dense cloud cover surrounding Venus. A complete description of the Magellan SAR imaging system is given in [5], so here we will only summarize the most important characteristics:

- Wavelength / frequency: 12.6cm (2.385 GHz - S band)
- Incidence Angle: 15° - 45° (nominal)
- Range resolution: 120m - 360m

- Azimuth resolution: 120m
- Pixel-spacing: 75m (full-resolution F-MIDR's)
- Number of looks: 5- 16

Figure 1 shows a 30km x 30km area imaged by Magellan. This area, which is located near (lat 30° N, lon 332° ), contains many small volcanoes. The volcanoes are illuminated from the lower-left side of the image. Observe that the larger volcanoes in this figure have the classic radar signature one would expect based on the topography; that is, the upward sloping surface of the volcano in near-range (close to the radar) scatters more energy back to the sensor than the surrounding flat plains and therefore appears bright. The downward sloping surface of the volcano in far-range (away from the radar) scatters energy away from the sensor and therefore appears dark. Together, these effects cause the volcano to appear as a left-to-right *bright-dark pair* within a circular planimetric outline. Near the center of the volcanoes, there is usually a summit pit that appears as a *dark-bright* pair because the radar energy reflects strongly from the far-range side of the summit pit back to the sensor, but not from near-range. For small pits, however, the resolution may not be high enough to separate the dark-bright pair.

The topography-induced features described above are the primary visual cues that geologists report using to locate volcanoes. However, there are a number of other, more subtle cues. The apparent brightness of an area in a radar image depends not only on the macroscopic topography but also on the surface roughness relative to the radar wavelength. Thus, if the flanks of a volcano have different roughness properties than the surrounding plains, the volcano may appear as a bright or dark (often circular) area instead of as a bright-dark pair. Volcanoes may also appear as radial flow patterns, texture differences, or disruptions of graben. (Graben are ridges or grooves in the planet surface that appear as bright "fracture" lines in the radar imagery.)

### 3 Ground Truth

In the volcano-location problem, as in many pattern recognition applications, real ground truth data may not exist. No one has ever been to Venus, and despite the fact that the Magellan data is the best imagery ever obtained of Venus, scientists cannot always determine with 100% certainty (due to factors such as image resolution, signal-to-noise level, etc. ) whether a particular image feature is indeed a volcano. Furthermore, for some domains there may not even be a precise definition of the objects of interest.

Although one could envision using unsupervised learning paradigms that try to discover structure in data, these methods are generally less well-developed and less reliable than the corresponding supervised techniques. Thus, to provide training data for a supervised system in domains where real ground truth does not exist, one must carefully design a scheme for collecting the training data in a manner that is consistent with the practical realities. This adds two important dimensions to the traditional classification learning problem:

1. Degrees of uncertainty need to be attached to the training examples; e.g., sure volcanoes vs. marginal ones.

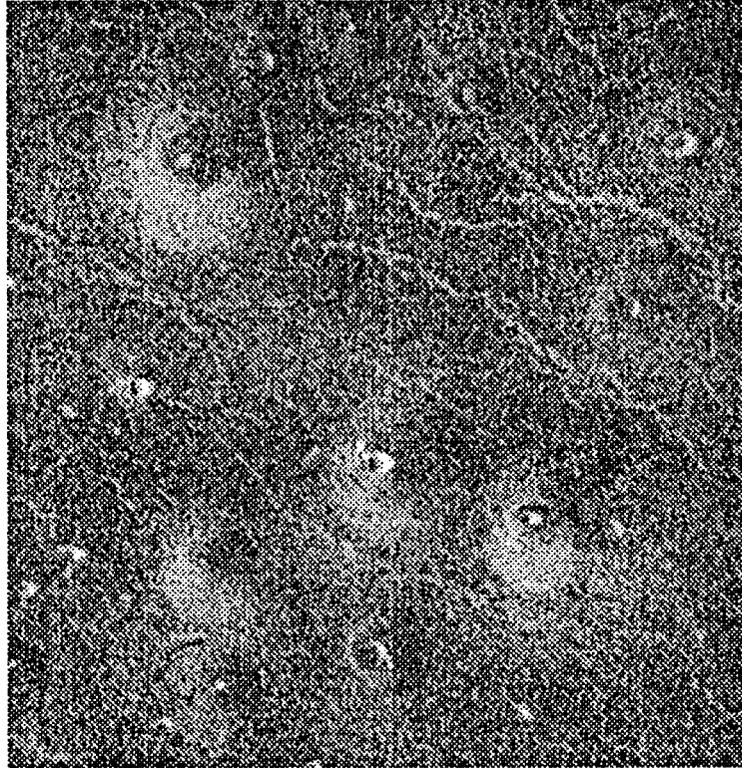


Figure 1: A 30km x 30km region from the Magellan SAR data, which contains a number of small volcanoes. The radar illumination is from the lower left, and the incidence angle is approximately 40°.

2. Disagreement among labelers must be taken into account when evaluating system performance; e.g., two scientists may disagree on a (possibly significant) subset of the data.

The issue of attaching uncertainty labels to the training examples is discussed in the next subsection (Section 3.1); the implications for evaluating system performance are considered in the succeeding subsection (Section 3.2).

### **3.1 Attaching Uncertainty Labels to the Training Examples**

To simplify the process of harvesting training data from the geologists, we have developed a software interface known as JARtool (JPL Adaptive Recognition Tool) [2]. JARtool permits the scientists to quickly identify volcanoes using a point and click graphical interface. The scientists can indicate size information by fitting circles (or other shapes) around each region of interest (ROI). The problem of labeling uncertainty is handled by asking the scientists to

give a subjective estimate of the probability that each object is actually a volcano. To avoid the pitfalls of biased subjective estimates, we allow the fewest necessary degrees of freedom in representing these probabilities. Through discussions with the scientists, we decided to quantize the confidence attached to each ROI into four levels:

**Category 1:** definitely a volcano, with all primary visual cues present; this represents a subjective probability in the range [0.95 -- 1.0].

**Category 2:** very likely to be a volcano, a non-essential visual cue is missing; subjective probability in the range [0.75 – 0.95].

**Category 3:** possibly a volcano, at least two of the primary cues are missing; subjective probability roughly in the range [0.5 – 0.7].

**Category 4:** only a pit is showing; likely to be a volcano but more evidence is needed.

The scientists believe that this quantization is both reasonable and appropriate. Note that Class 3 and Class 4 are not strictly ordered in the sense that a three is not necessarily more volcano-like than a four. They both indicate uncertain volcanoes, but the uncertainty is generally due to different factors. Figure 2 illustrates some typical volcanoes from each category.

### **3.2 Performance Evaluation with Uncertain Ground Truth**

- Given that the scientists cannot classify each object with 100% confidence, how can we assess how well our algorithms are performing? The basic idea is to measure the performance of individual scientists with respect to a “consensus ground truth”, where the consensus data is generated by several scientists working together discussing the merits of each candidate volcano (see Figure 3). The performance of an algorithm is considered to be satisfactory if, compared to consensus ground truth, its performance is as good as that of an individual scientist. The philosophy here is that if a single scientist is qualified to perform the analysis, then it is sufficient if our algorithms perform comparably.

We evaluated the individual performance of two scientists in the following way. Each was asked to label separately the volcanoes in a collection of four images using the subjective probability categories described above. Approximately one week later, the two scientists jointly labeled the same images starting anew, i.e., they did not directly use their previous individual labelings. Figure 4 shows the confusion matrices comparing each scientist’s labeling to the consensus.

The  $(i, j)$  entry in the confusion matrix is interpreted as the number of volcanoes labeled  $i$  by an individual scientist that were in fact labeled  $j$  in the consensus. ‘1’bus, scientist A labeled 5 volcanoes as category 2 that were listed in the consensus as category 1. The last row of the confusion matrix shows the number of volcanoes the scientist completely missed. The last column shows the number of false alarms by the scientist. Note that the  $(0, 0)$  entry is 0 by definition. The confusion matrices can also be shown graphically as in the plots at the right in Figure 4. Notice the significant number of off-diagonal elements. We will return to the scientists’ labeling performance in Section 6 when we evaluate the performance of our algorithms.

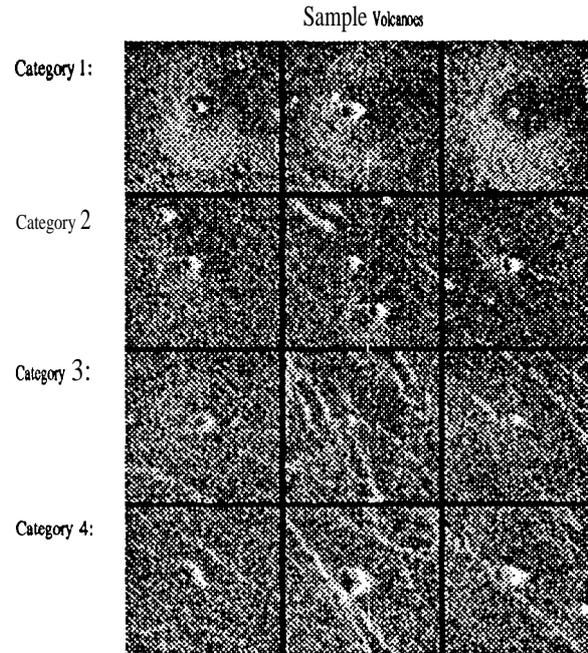


Figure 2: A small selection of ROIS as labeled by the geologists. The category 1 ROIs are believed by the geologists to almost certainly be true volcanoes. The category 2 ROIS are also very likely to be true volcanoes but one piece of evidence may be missing, e.g., a well-defined summit pit. The category 3 ROIs have some volcanic features but there is not enough evidence to say with confidence. The category 4's arc pits that appear to be volcanoes but evidence of a circular outline or bright-dark shading is absent.

## 4 Structure of the Algorithm

In this section, we provide an overview of the algorithm we have developed for finding small volcanoes on Venus. As illustrated in Figure 5, the algorithm operates in two distinct phases: (1) the learning phase and (2) the production phase. An overview of the processing in each phase is given here. The component blocks comprising each phase are described in detail in Section 5.

### 4.1 Learning Phase

The main function of the learning phase is to develop models of the objects of interest based on a number of examples and counter-examples. The learning phase should be generic enough so that the system may be applied to other detection tasks merely by supplying to it a new set of examples *without any need for reprogramming*. Three components in the system are constructed using learning from training data: the focus of attention (FOA) component, the feature learning (FL) component, and the classification learning (CL) component.

Generating the FOA component is the first processing step in the learning phase of the

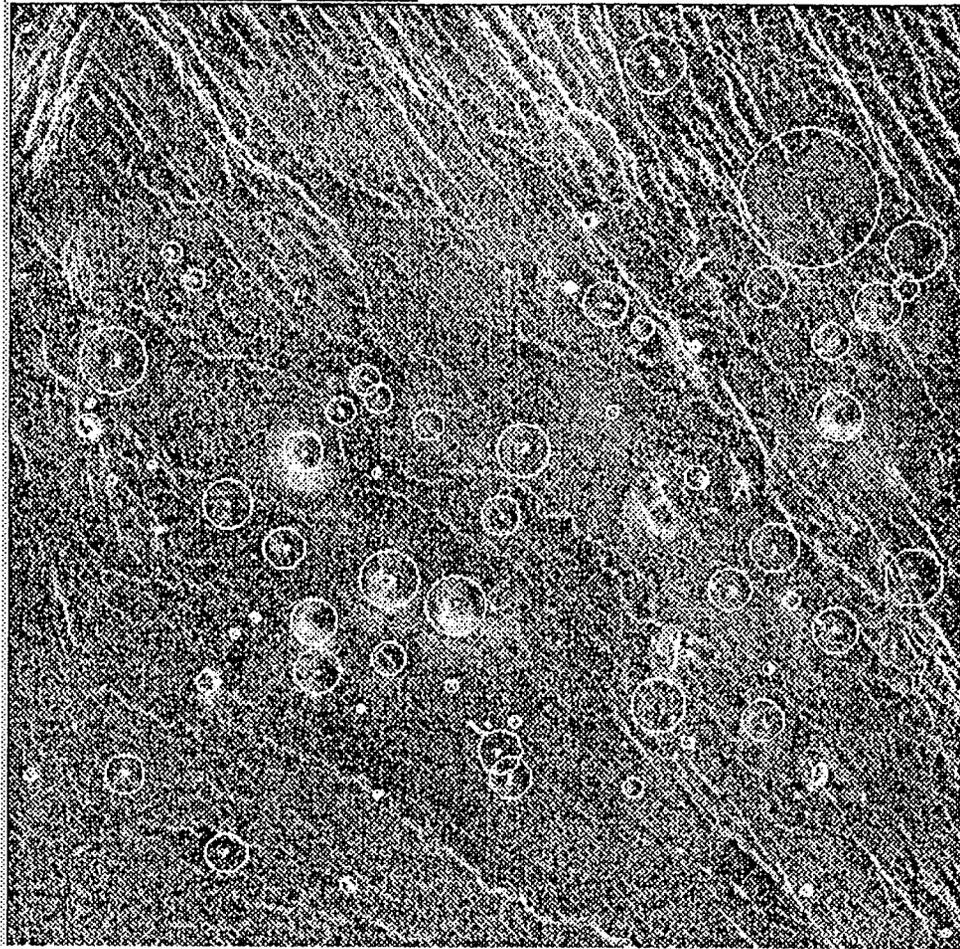


Figure 3: Magellan SAR image of Venus with consensus ground truth showing size and locations of small volcanoes.

system. The FOA is intended to select quickly candidate regions of interest with minimal computational effort. Use of the FOA eliminates the need to examine uninteresting areas in extensive detail. The candidate locations identified by the FOA are then checked against the consensus ground-truth list provided by the domain-experts. Based on comparison with the ground truth, the FOA candidates are separated into a set of examples (actual instances of the desired pattern) and counter-examples (candidates that do not in fact correspond to the object of interest — also called false alarms). The FOA is discussed in detail in Section 5.1.

The examples and counter-examples produced by the FOA algorithm are then processed by a feature learning algorithm (FL). This component attempts to derive a set of relevant features based on the examples and counter-examples. In our approach each feature is a linear combination of the pixel values of an image region. The vectors of combination coefficients (which may be seen either as ‘templates’ or as kernels of linear filters) are generated using principal component analysis of the examples (Section 5.2).

Measuring the feature values for a particular region of interest (ROI) is straightforward,

		Consensus						
		1	2	3	4	0		
A 07-93	1	2	8	1	1	6	1	0
	2	5	9	8	9	9		
	3	1	2	20	8	31		
	4	1	2	5	26	13		
	0	0	6	11	4	0		
		Consensus						
		1	2	3	4	0		
B 07-93	1	1	9	6	6	3	1	
	2	9	5	9	4	6		
	3	4	13	18	6	37		
	4	0	3	3	25	18		
	0	3	3	1	4	1	0	

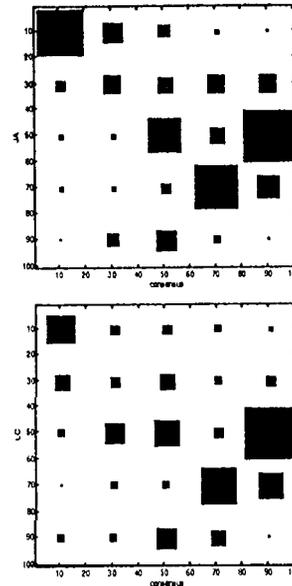


Figure 4: Single scientist performance compared to 'consensus' ground-truth. Labeling confusion matrices are shown for two scientists, A and B. The matrices are tabulated on the left hand side and are shown graphically on the right hand side. Notice the significant number of off-diagonal labels.

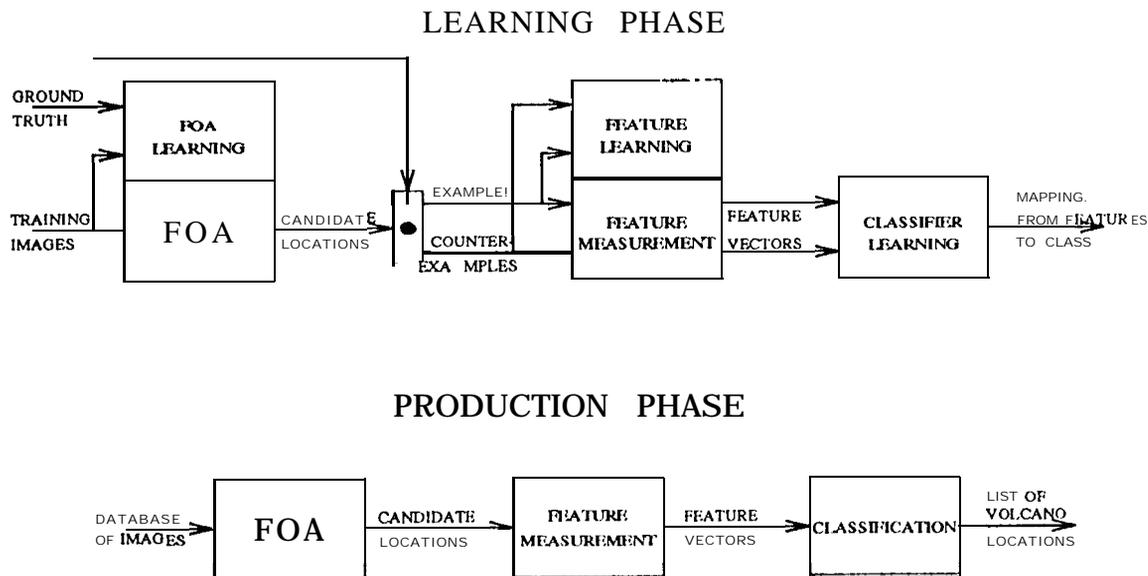


Figure 5: A block diagram showing the overall structure of our algorithm is illustrated here. The algorithm naturally separates into two distinct phases: a learning phase and a production phase. The main function of the learning phase is to develop models of the objects of interest based on examples and counter-examples. The production phase then applies these models to recognize objects of interest.

involving only a simple dot-product or cross-correlation of the ROI with each template, The

feature measurements map an ROI from a high-dimensional pixel space to a much lower dimensional feature space via projection. These lower dimensional feature vectors hopefully encode the essential or relevant information contained in each ROI.

The final stage of the learning phase is the classification learning (CL) component. The purpose of this component is to construct a mapping from the space of features to a class label. Based on the features associated with an ROI, the classifier declares the ROI to be either: (1) an actual object of interest or (2) a false alarm. The classifier will compensate for the FOA's tendency to respond to weak candidates (high false alarm rate). We use a quadratic classifier (also known as the Gaussian classifier) in the experiments that we present in later sections. We have experimented with a number of other well-established supervised learning methodologies including decision trees, neural networks, and kernel density estimation methods with similar results. The particular choice of classifier technology is not critical provided that the features are the right ones.

## 4.2 Production Phase

The production phase consists of examining a set of test images (or eventually a whole database) to find all instances of the objects of interest. Using the FOA component, each image is screened quickly to identify candidate locations of the objects of interest. For each candidate ROI, the feature templates learned by the FL box are used to measure feature values. These values are then passed to the classification algorithm, which assigns the ROI a class label (volcano vs non-volcano in the current application). Any ROIs designated by the algorithm to be legitimate objects of interest can then be cataloged in a database and made available for statistical and scientific analysis. (In some domains, human verification of the objects may be desirable. )

## 5 Components of the Algorithm

In the previous section, we outlined the basic structure of our algorithm at the level of detail shown in Figure 5. In this section, we will provide a more extensive description of each of the algorithm components. Section 5.1 describes both the learning and production phases of the focus of attention algorithm. Section 5.2 discusses feature learning and feature measurement. Classification is considered in Section 5.3.

### 5.1 Focus of Attention

The main goal of the FOA component is to detect as many volcanoes as possible in a quick and efficient pass. The input to the FOA are the full 'raw' images (eg. the SAR images of the surface of Venus); its output is in the form of a discrete set of fixed-size image neighborhoods containing an object of interest with reasonably high probability. The FOA is by design intended to be aggressive; i.e., it is acceptable for the FOA to generate a significant number of false alarms, as long as it misses very few actual targets. By using the FOA to identify candidate regions, more sophisticated approaches to feature extraction and classification are affordable, since the number of pixels under consideration is drastically reduced.

The constraints upon the FOA (speed and low miss-rate) suggest that a linear filtering operation should be used for this component. The problem of finding the best linear filter to detect a deterministic signal in white noise is well-known. The solution is the matched filter; i.e., the filter whose shape matches the shape of the signal one is trying to find. To find a random signal with non-zero mean in white noise, the filter should be matched to the mean of the signal.

Thus, for the volcano-location problem a matched filter is constructed by forming an average template from all the volcanoes in the training set. Before computing the average, however, we first normalize each volcano with respect to the local DC and local contrast level. Let  $\mathbf{v}_i$  denote the  $k \times k$  pixels in a window around the  $i$ -th volcano rearranged as a  $k^2$ -dimensional vector. Each region of interest in the training dataset is then normalized as follows:

$$\tilde{\mathbf{v}}_i = \frac{\mathbf{v}_i - \mu \cdot \mathbf{1}}{k\sigma} \quad (1)$$

where  $\mu$  is the local DC level (mean of the pixels in  $\mathbf{v}_i$ ),  $\sigma$  is the local contrast (standard deviation of the pixels in  $\mathbf{v}_i$ ), and  $\mathbf{1}$  is a  $k^2 \times 1$  vector of ones. The matched filter  $f$  is constructed simply by averaging the normalized examples. Although the resulting filter is guaranteed to have the zero DC property, we must renormalize the contrast to one.

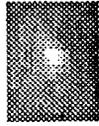
The matched filter response is computed as the normalized cross-correlation between  $f$  and each image patch (i.e., each image patch is normalized as in Equation 1 prior to computing the dot product with  $f$ ). The response is equivalent to the statistical cross-correlation between  $f$  and the image patch; thus, response values close to one indicate that the image patch is strongly correlated with the filter. Substantial savings in the computation of the cross-correlation can be achieved using separable kernel methods to approximate the 2-D kernel  $f$  as a sum of 1-D outer products.

The matched filter response image is thresholded at a level determined from training images. Nearby pixels that are above the threshold are clustered into a single candidate location. Results on a typical image are shown in Figure 7. The FOA candidates are displayed as boxes overlaid on image, while the ground truth locations are shown as circles. Although there are quite a few false alarms, recall that the goal of the FOA component is to achieve a low-miss rate while reducing the amount of data to be processed by later stages. The FOA algorithm is successful in detecting all the volcanoes from Categories 1 and 2, but misses a few from Categories 3 and 4.

We have experimented with several variations of the matched filter. One variation, which we call the size-binned matched filter, attempts to account for size information. With this approach, the training volcanoes are grouped into four size ranges based on the scientist-fitted diameters. A separate matched filter is constructed for each size range. The candidate locations identified by each matched filter are merged and consolidated into a master list of candidates.

We also considered a version of the matched filter that takes into account the subjective uncertainty of the scientists. This filter is constructed in the same way as the original FOA scheme, except each training volcano is multiplied by the scientist's subjective probability when averaging. Provided the subjective probabilities are unbiased, this method can be shown to have useful asymptotic properties [7].

Matched Filter Displayed as a Template



Matched Filter Displayed as a Surface Plot

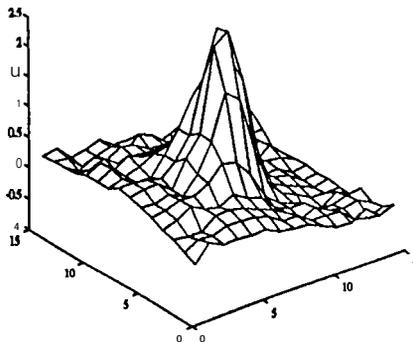


Figure 6: Left: The matched filter contains many of the features that planetary geologists report using when manually locating volcanoes. In particular, the matched filter encodes a bright central spot corresponding to a volcanic summit pit and the left-to-right bright-dark shading. Right: Response of the matched filter on the area shown in Figure 1. Bright points indicate a strong match — these will be selected as candidate locations.

## 5.2 Feature Synthesis

Determining features from examples is essential to developing a domain-independent algorithm. The feature learning component uses examples generated by the F<sub>OA</sub> algorithm to synthesize a number of features to be used for classification. As mentioned before, these examples are in the form of fixed-size image neighborhoods that we call regions of interest (ROIs). One must note right away that in the present application, generating features automatically is an underconstrained problem: the dimensionality of the space of all possible features is much too high relative to the number of examples that are available. The search must therefore be restricted to a family of potentially useful features; the choice of the family is of course essential to the success of the application. Given the unstructured nature of our data we have chosen to generate features using the classical technique of projecting the ROI onto a low dimensional subspace. Each feature is a coordinate in this space and may

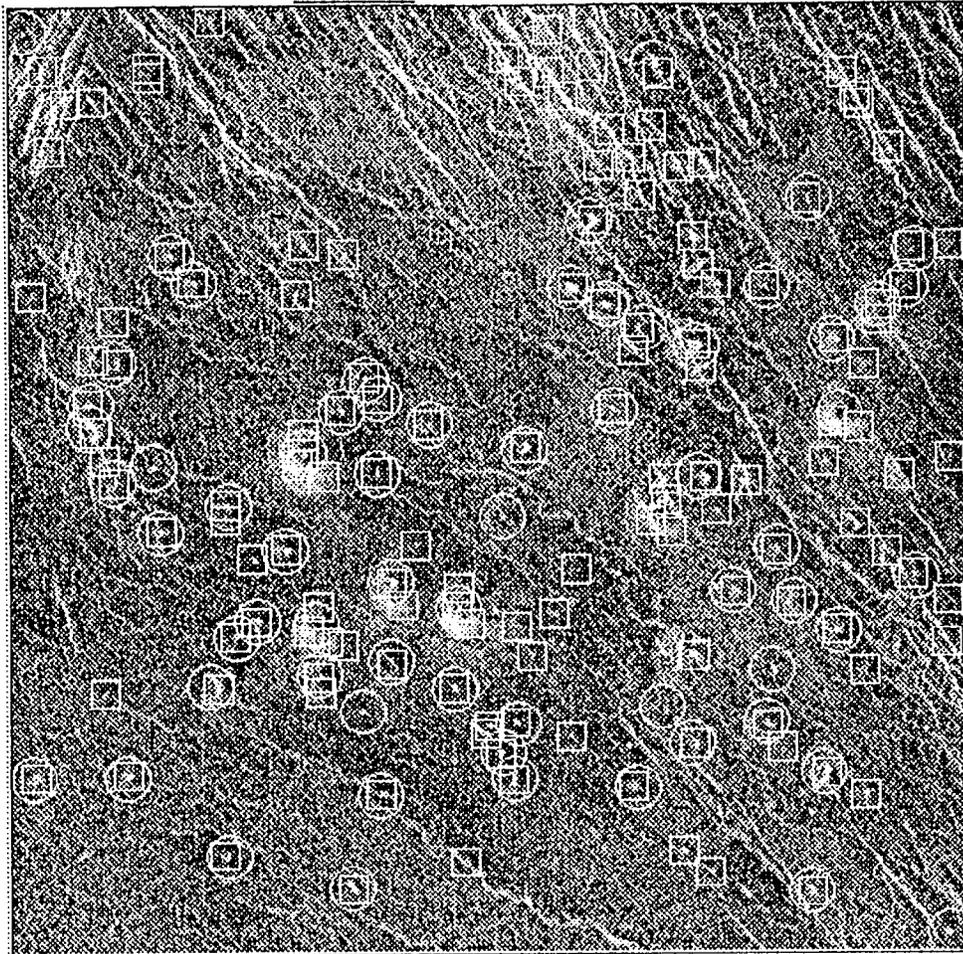


Figure 7: The output of the focus of attention algorithm on a typical image. Circles show the consensus ground truth volcano locations, while boxes show the candidate regions selected by the FOA. Thus, circles with boxes are detected volcanoes. Circles without boxes are missed volcanoes and boxes without circles are false alarms. Since the FOA acts as prescreening for other stages, the cost of a missed volcano is high compared to a false alarm.

be obtained as a linear combination of the ROI pixel values; this is equivalent to selecting an ROI-sized kernel for each feature and computing the inner product of the ROI with the kernel. Having selected this computational strategy, our problem is now one of determining the dimension  $q$  of the subspace on which to project and finding  $q$  basis vectors (i.e., the appropriate kernels) that span the subspace.

If one takes a probabilistic point of view, the Karhunen-Loève transform yields the 'best' subspace on which to project the data. This subspace is spanned by the highest-eigenvalue eigenvectors of the data covariance matrix. The full covariance matrix unfortunately cannot be computed reliably from the number of examples we typically have available. However, since we are only interested in the highest-eigenvalue eigenvectors, the approximate K-L

basis vectors can be found by the method of principal components. This technique has been used already in visual pattern recognition, for example, by Turk and Pentland for face recognition [8].

Despite its intuitive appeal, there are a number of arguments against using such a simple image-based approach for recognition: most notably, it is not invariant with respect to translation, rotation, scaling and direction of illumination. For the volcano problem these limitations are not critical since the FOA algorithm ‘centers’ the volcanoes well, and the volcanoes have a high degree of rotational symmetry. However, for the general problem these invariance issues must be addressed systematically and eventually resolved.

The principal components can be determined as follows. Each normalized example is placed as a column in an  $n \times m$  matrix  $X$ , where  $n$  is the number of pixels in each ROI and  $m$  the number of examples (ROIs). Using the singular value decomposition, we can factor  $X$  as follows:

$$X = USV^T \quad (2)$$

For notational convenience, we will assume  $m$  is less than  $n$  since this is usually the case. Then in Equation 2,  $U$  is an  $n \times m$  matrix such that  $U^T U = I_{m \times m}$ ,  $S$  is  $m \times m$  and diagonal with the elements on the diagonal (the singular values) in descending order, and  $V$  is  $m \times m$  with  $V^T V = V V^T = I_{m \times m}$ . Notice that any column of  $X$  (equivalently, any ROI) can be written exactly as a linear combination of the columns of  $U$ . Furthermore, if the singular values decay quickly enough, then the columns of  $X$  can be very closely approximated using only linear combinations of the first few columns of  $U$ . That is, the first few columns of  $U$  serve as an approximate basis for the entire set of examples in  $X$ .

The columns of  $U$  are, in fact, the principal component vectors which will serve in place of the K-L vectors. These are shown in Figure 8 reshaped into ROIs. Notice that the first ten or so exhibit structure, while the remainder appear very random. This suggests that we should use a subspace of dimension  $q = 10$ . The singular values plotted in the figure indicate that 6 – 10 features capture most of the information in the examples.

Ideally, we would like to describe objects of interest using as few features as necessary. Having a small number of features relative to the number of training examples improves the likelihood that the classifier will not overfit to the training data. We expect that a certain number of features will be required in order to describe the natural variability of an object. Any additional variability of the object due to spatial shifting, scaling, rotation, or noise will increase the number of features required to produce a good representation. One might mistakenly make the following argument: shifting has two degrees of freedom, scaling has one d.o.f., and rotation has one d.o.f., therefore, encoding these variations should only require four additional features. While this might be true if we were using very non-linear features, it is definitely not true for the case where the features are restricted to be linear functions of the pixels. Thus, the number of principal components needed can increase dramatically in an attempt to encode these additional non-linear parameters. We are currently working on a method to “undo” or normalize with respect to these variations, but it is still in the research stage. Fortunately, as we have already remarked, these issues are not critical for the volcanoes problem,

Once we have determined the appropriate number  $q$  of features to use, we can calculate

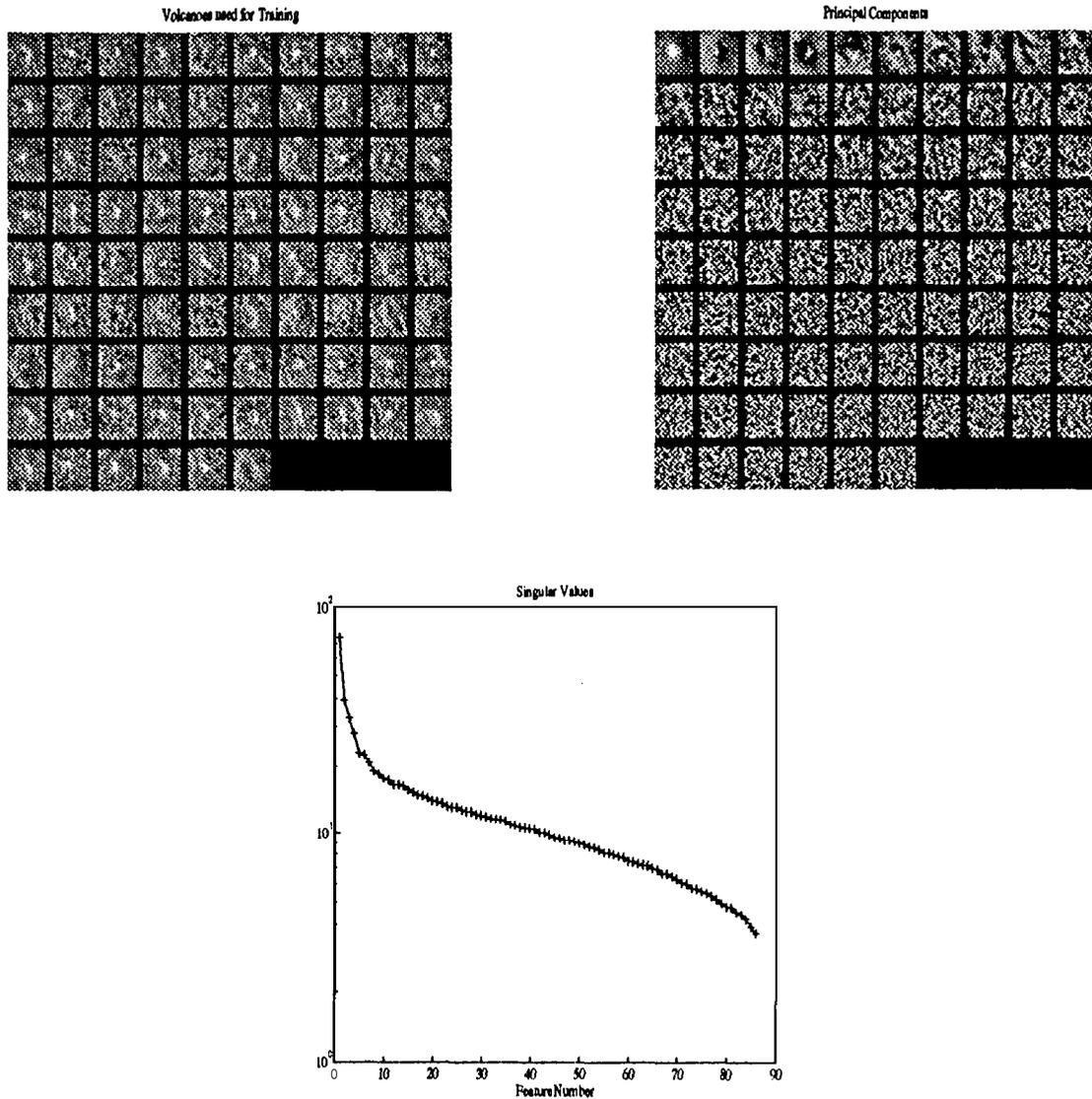


Figure 8: Upper left shows collection of volcanoes used for feature synthesis. Upper right shows principal components derived from the examples. Notice that the first 10 or so exhibit structure while others look like random noise. Bottom: Singular values indicate importance of each of the features for representing the examples. Rapid fall-off of singular values shows that examples can be closely approximated using only a few principal components.

the feature values for an ROI as follows:

$$\mathbf{y} = \left[ \mathbf{u}_1 \ \mathbf{u}_2 \ \dots \ \mathbf{u}_q \right]^T \mathbf{x} \quad (3)$$

where  $\mathbf{x}$  is the ROI reshaped as an  $n$ -dimensional vector of pixels and  $\mathbf{y}$  is a  $q$ -dimensional vector of features. The feature vectors  $\mathbf{y}$  serve as the input to the classification algorithm.

A final point to note about the feature learning algorithm is that like the FOA algorithm, the scientist's subjective probabilities can be taken into account during training. Since the

principal components are the eigenvectors of the sample covariance matrix, the subjective probabilities are used to calculate a modified covariance matrix as follows:

$$\Sigma_p = \frac{\sum_{i=1}^m p_i \mathbf{x}_i \mathbf{x}_i^T}{\sum_{i=1}^m p_i} \quad (4)$$

where  $p_i$  is the probability that example  $\mathbf{x}_i$  corresponds to a true volcano. Note that this formula also encompasses the deterministic labeling method in which each  $p_i$  is simply equal to 1. The probabilities can be taken into account directly in the matrix  $X$  by multiplying column  $i$  by  $\sqrt{p_i/p}$  where  $p$  is the denominator of Equation 4.

### 5.3 Classification

The problem of classifier training in supervised learning problems has been well-studied. As we commented earlier, we believe that if the feature learning algorithm is working well, then the choice of classifier is not critical. We used a quadratic (Gaussian) classifier for the experiments reported in the next section, but we have also used decision trees, kernel density estimation, etc. and obtained similar results.

Briefly, the quadratic classifier is the optimal classifier (in the Bayes sense) if the class-conditional probability densities of the feature vector  $\mathbf{y}$  are multivariate Gaussian, i.e., if

$$\begin{aligned} p(\mathbf{y}|\omega_1) &\sim N(\mu_1, \Sigma_1) \\ p(\mathbf{y}|\omega_2) &\sim N(\mu_2, \Sigma_2) \end{aligned} \quad (5)$$

Even if the Gaussian assumption does not strictly hold, the quadratic classifier usually performs well provided the densities are unimodal. There are two common interpretations of the algorithm, both of which are useful. The first interpretation is that the algorithm estimates posterior probability densities of the two classes given the feature vector  $\mathbf{y}$ . For example, the class 1 posterior probability is estimated as:

$$p(\omega_1|\mathbf{y}) = \frac{p(\mathbf{y}|\omega_1)p(\omega_1)}{p(\mathbf{y}|\omega_1)p(\omega_1) + p(\mathbf{y}|\omega_2)p(\omega_2)} \quad (6)$$

where  $p(\omega_1)$  and  $p(\omega_2)$  are the prior probabilities for each class. Training the quadratic classifier is particularly easy, since the class-conditional densities are completely determined from the class mean and covariance matrix. Thus, we simply have to estimate these parameters from the training data.

The other common interpretation of the quadratic classifier is as a nearest distance classifier, where the distance metric is the Mahalanobis distance:

$$d_i(\mathbf{y}) = (\mathbf{y} - \mu_i)^T \Sigma_i^{-1} (\mathbf{y} - \mu_i) \quad (7)$$

An unknown sample  $\mathbf{y}$  is assigned to the class  $i$  for which  $d_i(\mathbf{y})$  is the smallest. As seen from Equation 7, the minimum distance rule corresponds to using a quadratic hyper-surface to

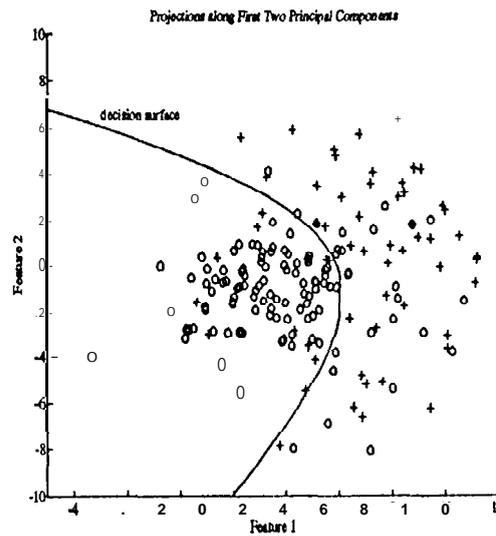


Figure 9: Illustration of the quadratic classifier in two dimensions. Volcanoes (+) and non-volcanoes (o) are mapped from pixel space to a two-dimensional feature space. The quadratic decision surface attempts to separate the samples in the feature space. Improved separation can be obtained using a higher dimensional feature space. (We have found that using 6-10 dimensions works best.)

separate the two classes. A two-dimensional example illustrating the quadratic separating surface is shown in Figure 9. The plotted points are the first two SVD features measured on the ROIS shown in Figure 7. Although the quadratic decision surface does a reasonable job of separating the points, the results are much better if the ROIS are mapped to a six-dimensional space.

Both implementations of the algorithm can be adjusted to be more or less aggressive. Since in the first interpretation the algorithm declares class  $\omega_1$  if the posterior probability of this class exceeds a threshold, the algorithm can be made more aggressive by lowering the threshold. By varying this threshold we can make the algorithm more or less aggressive in declaring objects to be of class  $\omega_1$ . For the second version, one can simply imagine moving the separating hyper-surface closer to one class or the other to change the operating point.

## 6 Experimental Performance Results

In this section, we present the experimental results obtained using our algorithm to locate small volcanoes in Magellan SAR imagery. The performance of the algorithm in the volcano-location task is shown to be comparable to the performance of individual scientists.

The experiments were performed on four 1024 x 1024 images using a cross-validation paradigm. That is, three images were used to train the algorithm, while the fourth image was reserved for testing. This process was repeated four times with each image serving as the test image. All results were scored relative to the consensus ground truth treating the

subjective labels 1, 2, 3, and 4 as true volcanoes.

The figure of merit that we measure is the number of detections (true volcanoes identified) versus the number of false alarms (non-volcanoes mistakenly identified as volcanoes) summed over the four test images. In all the performance plots, detection and false alarm rates are expressed *as a percentage relative to the number of true volcanoes (163) in the four images*. Thus, the detection rate is bounded above by 100%, while the false alarm rate can be arbitrarily large. The use of percentages allows one to easily compare results across different images and across different labeling strategies. For most of the algorithm variations we have considered, there is a parameter that can be adjusted to make the algorithm more or less aggressive in declaring volcanoes. Varying this parameter generates a curve comparable to a standard ROC (receiver operating characteristic) curve.

As a basis for comparison (see Section 3.2), we evaluated the labeling performance of three scientists, who are all familiar with the Magellan data and with the appearance of volcanoes in the data. The DFA point for each scientist is shown as a labeled asterisk in Figure 10. Note that the  $x$ -axis is logarithmic.

Figure 10 also shows the performance of the matched filter focus of attention algorithm (solid line). The matched filter algorithm has one free parameter (a threshold), which can be adjusted to make the algorithm more or less aggressive in declaring volcanoes. Since the purpose of the 170A is to provide a low rate of missed volcanoes, the threshold was selected to make the algorithm operate at the point ( $f \approx 300\%$ ,  $d \approx 90\%$ ). At this operating point, the matched filter has the following highly desirable feature: it detects all the Category 1 and Category 2 volcanoes. The only misses are of Category 3 and Category 4 volcanoes.

The performance of the size-binned matched filtering scheme of Section 5.1 was also evaluated. The idea behind this experiment was to examine the effect of ignoring scale information in the original single-scale matched filter scheme. For the size-binned algorithm however, it is difficult to obtain a performance *curve* because there are multiple thresholds — one for each size matched filter; nevertheless, the performance for several threshold combinations can easily be determined. The corresponding DFA points are shown in Figure 10 with  $\mathbf{x}$ 's. Observe that the size-binned matched filter improves only slightly upon the performance of the single-scale matched filter. The size-binned algorithm, however, requires considerably more computation time, is more difficult to synthesize, and has more parameters to adjust than the single-scale version. Hence, we continue to use the single-scale matched filter as the baseline focus of attention algorithm.

Like the size-binned matched filter, the probability-weighted matched filter did not yield any improvement in performance over the original matched filter. In fact, the probability-weighted filter performed slightly worse,

Next we evaluated the end-to-end performance of the baseline algorithm. Recall that each candidate ROI is mapped from pixel space to feature space by measuring the values of the features that were learned by the FL stage during training. Candidates are then classified as volcano or not-volcano by the quadratic (Gaussian) classifier of Section 5.3.

The overall performance of the baseline algorithm is presented in Figure 11. The performance is shown for two cases: (1) six features and (2) two features. The performance with six features is approaching that of the scientists for this task. *In particular, the algorithm detection rate is clearly within 10% of Scientist B's detection rate at the same false alarm*

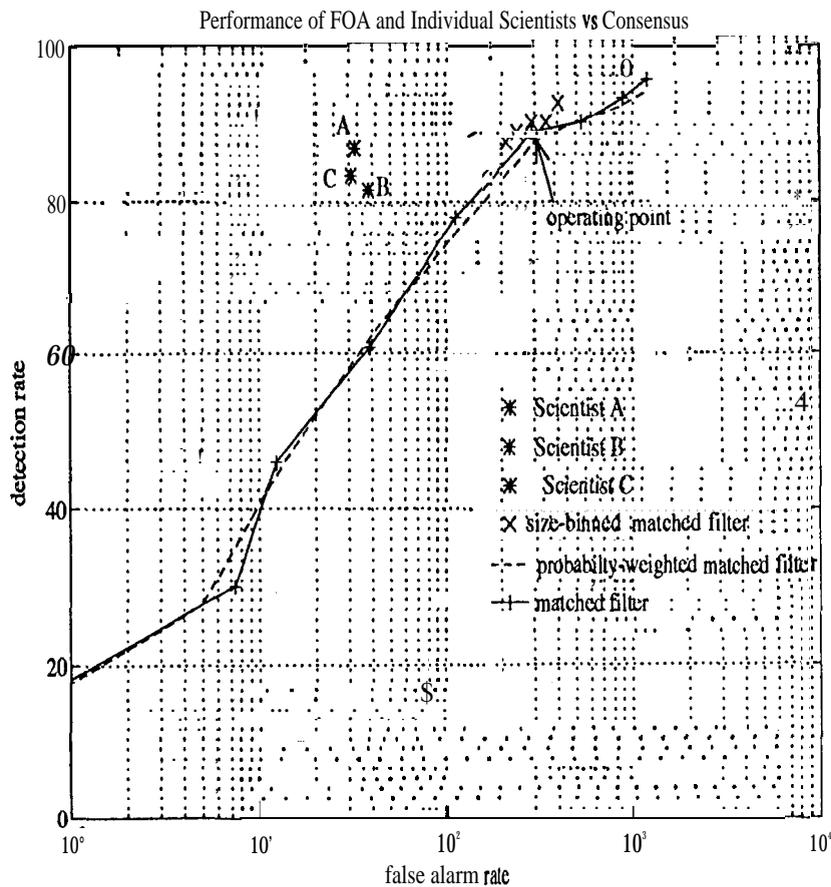


Figure 10: The asterisks show the labeling performance of three scientists, who are all familiar with the Magellan data and with the appearance of volcanoes in the data. Their DFA points define the performance level we are trying to achieve with our algorithms. The focus of attention algorithm is shown as the solid line. The FOA threshold was selected to establish an operating point with a low rate of missed volcanoes. At this operating point the FOA detects *all* Category 1 and Category 2 volcanoes. Two variants of the matched filter were also studied — a size-binned matched filter bank and a weighted matched filter constructed using the scientists subjective probability labels.

rate.

Also shown on Figure 11 is the FOA curve. One might be curious as to whether the whole volcano labeling task could have been performed using the matched filter (as in [11]). This figure clearly shows, however, that the FOA alone is significantly worse than the combination of an aggressive FOA followed by classification. This raises an interesting question. Since the features learned by the FL stage are linear filters, these features could be computed at *every* pixel in an image simply by convolution. The classification algorithm could then attempt to classify each pixel as volcano or non-volcano. How would the performance of the classifier-only algorithm compare to the FOA-classifier combination? If there is an

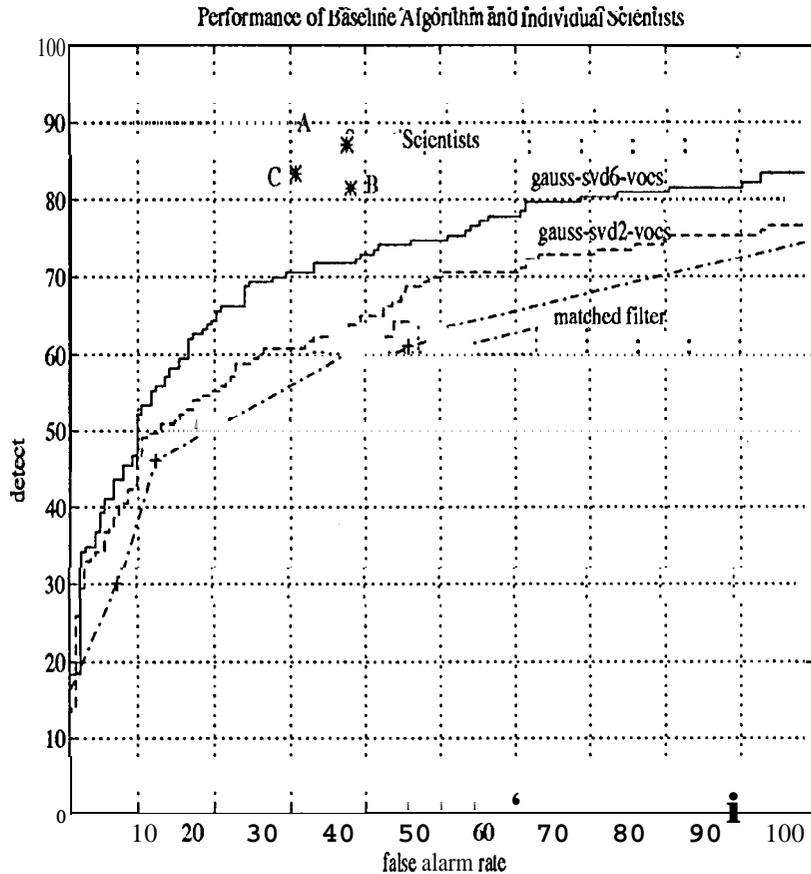


Figure 11: Performance of the overall algorithm compared to individual scientists. Using the Gaussian classifier with 6 SVD features yields performance that is close to that of the scientists. The performance using just 2 SVD features is somewhat worse, but still better than the matched filter alone.

improvement in performance, would it be substantial enough to offset the significant increase in computational costs? We intend to address these questions in future research.

Another question that arises is whether the detection vs false alarm rate (DFA) curves can be parameterized in some manner that will provide summary statistics by which different algorithms can be compared. We have derived the result that under moderate assumptions, the DFA curves can be expressed as a function of two parameters  $r$  and  $\nu$ . The key assumption is that an algorithm makes decisions based on a scalar quantity (e.g., the log-likelihood discriminant function), which is class-conditionally Gaussian distributed. Designating  $(\mu_1, \sigma_1)$  and  $(\mu_2, \sigma_2)$  as the class-conditional Gaussian parameters for volcanoes and non-volcanoes respectively, we can define  $r$  and  $\nu$  as follows:

$$\nu = \frac{\mu_1 - \mu_2}{\sigma} \quad (8)$$

$$r = \frac{\sigma_1}{\sigma_2} \quad (9)$$

The theoretical DFA function is then given approximately by:

$$d \approx \frac{100 \frac{N_1}{K}}{1 + \left( e^{-\alpha \nu} \cdot \left( \frac{100 N_2}{K f} - 1 \right) \right)^{\frac{1}{r}}} \quad (10)$$

where  $d$  is the detection rate,  $f$  is the false alarm rate (both expressed in the usual way as percentages relative to the total number of volcanoes  $K$ ),  $\alpha$  is a constant approximately equal to  $2.4/\sqrt{2}$ , and  $N_1, N_2$  are the number of volcanoes and false alarms out of the FOA. We have found that Equation 10 provides an excellent fit to our experimentally-obtained DFA curves. The best-fit values of  $\nu$  and  $r$  can be used to characterize the empirical DFA curves more compactly, which is useful when comparing many DFA curves. We now use this method in an empirical study to determine how many SVD features give the best detection performance.

Figure 11 indicates that using six SVD features provides better classification than two features. This result was to be expected based on the singular value decay curve shown in Figure 8. Clearly, more than two of the singular values have significant energy; the right number of features appears to be between six and ten. We performed an experiment to determine empirically how many SVD features would yield the best performance. The results are shown in two different ways. Figure 12-left shows the measured detection rate versus the number of SVD components at a few selected false alarm rates. Figure 12-right shows the fitted DFA performance parameters  $\nu$  and  $r$ . Notice that  $\nu$  (defined in Equation 8) is relatively flat across the whole range of svd number, although it is somewhat better between 4 and 13. The  $r$  parameter is also better (in this case smaller), between 4 and 13. Since both the detection curves and fitted-parameters are quite flat over range of svd features, we conclude that the performance is relatively insensitive to the exact number of features, as long as at least four are used. Beyond ten features, the classifier may recognize that the features are very noisy and don't carry much information, which explains the flatness of the curves versus number of features. However, if too many noisy features are included, the performance begins to degrade.

We have performed a preliminary experiment to determine whether using the scientists subjective labels in feature learning would yield an improvement in performance. The results shown in Figure 13 indicate that the performance actually degraded slightly. (We also found the same result when using the probabilities for FOA). There are several possible explanations that we are investigating. The subjective probabilities stated by the scientists may be uncalibrated. We are examining whether these probabilities can be determined based on labeling experiments. When scoring the algorithms, we treat all categories in the consensus as true volcanoes; however, in training the weighting emphasizes features that are useful for category 1 and category 2. The end result may be that we do better classifying the good volcanoes but worse on the marginal ones. Finally, small sample size may be a factor contributing to the degraded performance.

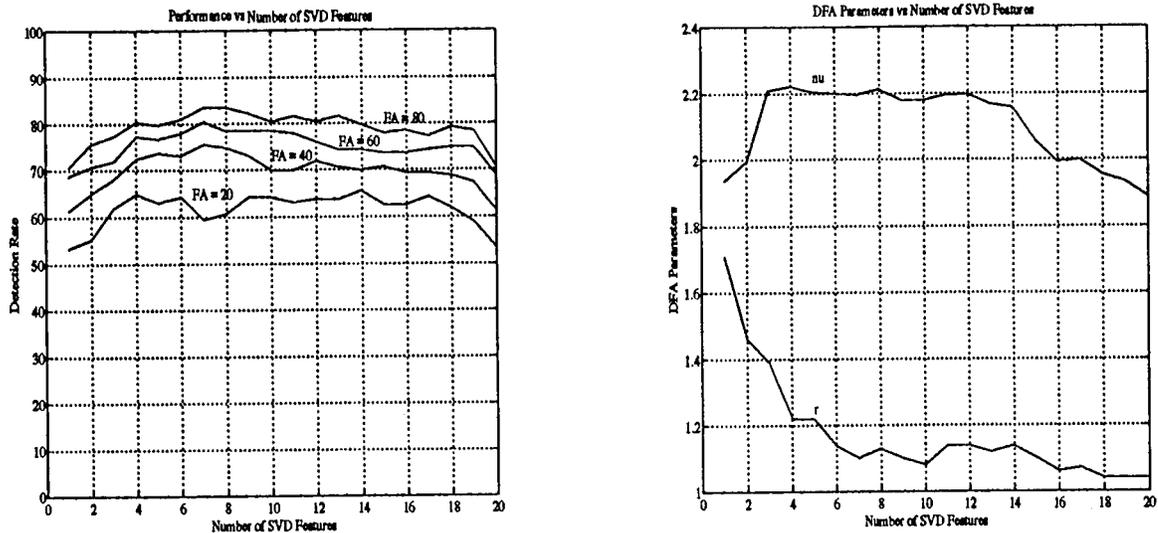


Figure 12: Performance versus number of SVD features. Left graph is empirically measured performance showing detection rates vs number of features at a few selected false alarm rates. Second graph shows parameters fitted to DFA curves as a function of the number of features used. Note that big  $\nu$  and small  $r$  corresponds to the best performance.

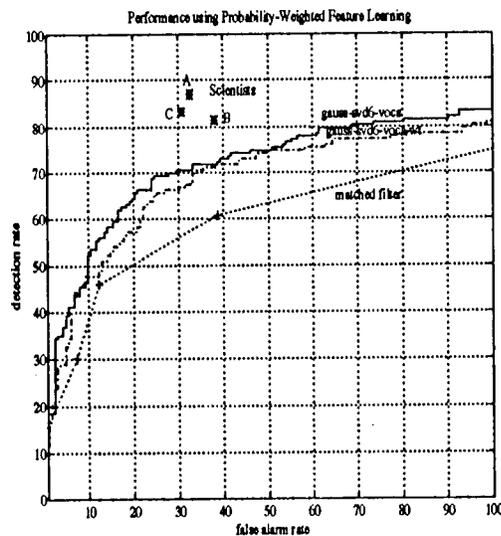


Figure 13: Performance of the SVD algorithm when features are learned using probabilistic weightings determined by the scientists' subjective labels. The performance is not as good as for the deterministic strategy; refer to the discussion in the text.

## 7 Discussion and conclusion

We have developed system for locating patterns in geographical images, The system, which is based on the classical techniques of linear filtering and supervised classification, is trained

by clicking a mouse on examples of the objects of interest. The system is being applied to the problem of locating small volcanoes in the Magellan SAR imagery of Venus. Tests conducted on four images containing  $\sim 160$  volcanoes show that our algorithm is performing nearly as well as trained human observers.

To summarize the results of our research so far:

1. A simple feature extraction scheme based on principal component analysis of training data is successful in a visual pattern recognition application with relatively featureless geographical patterns.
2. The number of features required to achieve best performance is small, around 6.
3. Considerable speedup of computations may be achieved using a 1-filter region-of-interest detector.
4. The training examples supplied by human experts may be quite unreliable; we have presented a methodology for measuring and for overcoming this problem.
5. Classification based on features may be performed by a cheap and well-understood Gaussian classifier, as opposed to more co-replicated perception; and decision trees.

The ultimate goal of our research is to develop a system that can easily be ported to new visual pattern recognition problems merely by supplying it with a new set of examples. Despite our initial success on the volcano problem, we believe that a number of issues remain to be solved before we can generalize to substantially different dataset types. Achieving invariance to translation, scaling, rotation, and illumination without renouncing to the advantages of filter-based processing is fundamental. Making use of local 'feature'-type information, making use of counter-examples, and allowing the scientists to enter "hints", e.g. find this object at any scale, are other open-issues one would like to resolve and incorporate into a domain-independent pattern recognition system.

## References

- [1] J.C. Aubele and E.N. Slyuta. Small domes on venus: characteristics and origins. *Earth, Moon, and Planets*, 50/51:493-532, 1990.
- [2] M.P. Burl, U.M. Fayyad, J.L. Loch, P. Smyth, and J.S. Yu. *JPL Adaptive Recognition Tool (JARtool)*. Jet Propulsion Laboratory - NASA, 0.9 edition, February 1993.
- [3] U.M. Fayyad and K.B. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proc. of the Thirteenth Inter. Joint Conf. on Artificial Intelligence*, Chambery, France, 1993.
- [4] U.M. Fayyad, N. Weir, R. Roden, S. Djorgovski, and R. Doyle. Using machine learning techniques to automate sky survey catalog generation. In *Proc. of 6th Space Operations, Applications and Research Symposium*. Morgan Kaufman, Houston, TX, 1992.

- [5] U.M. Fayyad, N. Weirand, and S. Djorgovski. Skicat: A machine learning system for the automated cataloging of large-scale sky surveys, In *Proc. of the Tenth International Conference on Machine Learning*, 1993.
- [6] G.H. Pettengill, P.G. Ford, W.T.K. Johnson, R.K. Raney, and L.A. Soderblom. Magellan: radar performance and data products. *Science*, 252:260-265, 1991.
- [7] P. Smyth. Learning with probabilistic supervision. forthcoming.
- [8] M. Turk and A. Pentland. Eigenfaces for recognition. *J. of Cognitive Neurosci.*, 3:71-86, 1991.
- [9] U. M. Fayyad. *On the Induction of Decision Trees for Multiple Concept Learning*. PhD thesis, Department of Electrical Engineering & Computer Science, The University of Michigan, Ann Arbor, 1991.
- [10] N. Weir, S. Djorgovski, and U.M. Fayyad. Skicat: A system for the scientific analysis of the palomar-stsci digital sky survey. In *Proc. of Astronomy from Large Databases II*, page 509, Munich, Germany: European Southern Observatory, 1993.
- [11] C.R. Wiles and M.R.B. Forshaw. Recognition of volcanoes on venus using correlation methods. *Image and Vision Computing*, 11(4):188-196, 1993.