Comparison of GENIE and Conventional Supervised Classifiers for Multispectral Image Feature Extraction

Neal R. Harvey, James Theiler, Steven P. Brumby, Simon Perkins, John J. Szymanski,
Jeffrey J. Bloch, Reid B. Porter, Mark Galassi, A. Cody Young

Abstract— **We have developed an automated feature detection/classification system, called** Genie **(GENetic Imagery Exploitation), which has been designed to generate image processing pipelines for a variety of feature detection/classification tasks.** Genie **is a hybrid evolutionary algorithm that addresses the general problem of finding features of interest in multi-spectral remotely-sensed images. We describe our system in detail together with experiments involving comparisons of** Genie **with several conventional supervised classification techniques, for a number of classification tasks using multi-spectral remotely-sensed imagery.**

Keywords— **Supervised Classification, Genetic Programming, Image Processing, Evolutionary Algorithms, Multi-spectral Imagery, Remote Sensing.**

I. Introduction

LARGE volumes of remotely-sensed multi-spectral data are being generated from an increasing number of increasingly sophisticated airborne and spaceborne sensor systems. While there is no substitute for a trained analyst, exploitation of this data on a large scale requires the automated extraction of specific features of interest. Creation and development of task-specific feature-detection algorithms is important, yet can be extremely expensive, often requiring a significant investment of time and effort by highly skilled personnel.

Our particular interest is the pixel-by-pixel classification of multi-spectral remotely-sensed images, not only to locate and identify but also to delineate particular features of interest. These range from broad-area features such as forest and open water to man-made features such as buildings and roads. The large number of features in which we are interested, together with the variety of instruments with which we work, make the hand-coding of suitable feature-detection algorithms impractical. We are therefore using a supervised learning approach that can, using only a few hand-classified training images, generate image processing pipelines that are capable of distinguishing features of interest from the background. We remark that our approach is to consider the two-class problem: although many applications require the segmentation of an image into a larger number of distinct land-cover types, we consider the simpler problem of identifying a single class against a background of "other" classes.

In applying general-purpose supervised learning tech-

niques to multi-spectral imagery, the usual approach is to employ purely spectral input vectors, formed by the set of intensity values in each spectral channel for each pixel in the image. These vectors provide a convenient fixed-dimensionality space in which conventional classifiers can often work well. It is clear, however, that spatial relationships (such as texture, proximity, or shape, all of which are disregarded with purely spectral vectors) can be very informative in scene classification. Many different kinds of extra spatial context information could be added to the spectral information, as additional dimensions of the pixel input vector. The problem is that there exists a combinatorially vast choice for these additional vector dimensions; yet it is clear that a suitable choice of additional dimensions could make classification much easier. Unfortunately, this suitable choice is, in general, application-specific.

To address this problem, we have developed a hybrid evolutionary algorithm called Genie (GENetic Imagery Exploitation) [2], [3], [4], [5], [6], [7], [8], that searches through the space of image processing algorithms. Genie is a hybrid in that the evolutionary part of the program attempts to identify a pipeline of image processing operations which transform the raw multi-spectral data planes into a new set of image planes; these intermediate "scratch" planes are then used as input to a conventional supervised classification technique to provide the final classification results.

When adopting an evolutionary approach, a critical issue is the representation of candidate solutions in order that they may be effectively manipulated. We use a genetic programming (GP) method of representation of solutions, due to the fact that each individual will represent a possible image processing algorithm. GP has previously been applied to image-processing problems, including: edge detection [9], film restoration [10], face recognition [11] and image segmentation [12]. The work of Daida et al. [13] and Bandyopadhyay and Pal [14] (as well as our own work, cited above) is of particular relevance since it demonstrates that GP can be employed to successfully evolve algorithms for real tasks in remote-sensing applications.

The beauty of an evolutionary approach is its flexibility: all that is required is a representation for candidate solutions, a fitness measure for comparing candidate solutions, and a scheme for "mutating" candidate solutions into other candidate solutions. Many varied problems beyond image processing have been successfully solved using evolutionary computation, from optimizing of dynamic routing

in telecommunications networks [15] to designing protein sequences with desired structures [16], and many others.

This paper describes our system in detail together with experiments involving comparisons of GENIE with several conventional supervised classification techniques, for a number of classification tasks using multi-spectral remotely-sensed imagery.

The remainder of the paper is organized as follows: Section II describes the GENIE system in detail. Section III describes the conventional supervised classification techniques with which GENIE is to be compared. Section IV describes the data and classification tasks on which the algorithms are to be tested and compared. Section V describes the results of the comparisons. Section VI describes further comparison with multi-class versions of the supervised classifiers. Finally, section VII discusses these results and concludes.

II. THE GENIE SYSTEM

GENIE employs a classic evolutionary paradigm: a population is maintained of candidate solutions (*chromosomes*), each composed of interchangeable parts (*genes*), and each assessed and assigned a scalar fitness value, based on how well it performs the desired task. After fitness determination, the evolutionary operators of selection, crossover and mutation are applied to the population and the entire process of fitness evaluation, selection, crossover and mutation is iterated until some stopping condition is satisfied.

A. Training Data

The environment for each individual in the population consists of *data* planes, each of these planes corresponding to a separate spectral channel in the original image, together with a *weight* plane and a *truth* plane. The weight plane identifies those pixels to be used in training – these are the pixels for which the analyst is confident in identifying as either "true" and "false": *true* defines areas where the feature of interest exists; *false* defines areas where that feature does not exist. The actual delineation of true and false pixels is given by the truth plane. This arrangement permits us the flexibility (not used in this study) to employ both real-valued weights (representing degrees of confidence or of importance) and real-valued truth (corresponding to retrieval of continuous valued properties). The data in the weight and truth planes may be derived from actual ground truth (collected on the ground, at or near the time the image was taken) or from the best judgement of an analyst looking at the data. Because collecting ground truth data is so expensive, our system employs a graphical interface called ALADDIN to assist the analyst in making judgements about and marking out features in the data. The analyst can view a multi-spectral image in a variety of ways, and can create training data by painting directly on the image using a computer mouse. Fig. 1 shows an image alongside the markup that an analyst provides as "ground truth." Figs. 4(b), and 6(b) show further examples where the analyst has marked out the desired feature on the image.

B. Encoding Candidate Solutions

Each individual *chromosome* in the population consists of a fixed-length string of *genes*. Each gene in GENIE corresponds to a primitive image processing operation. Therefore the entire chromosome describes an algorithm consisting of a sequence of primitive image processing operations.

A single gene consists of an operator name, a list of input planes, specifying from which plane input is to come; a list of (usually one) output plane; and a list of scalar parameters. Parameters may be integer, floating point, or categorical. Each gene used in GENIE takes one or more distinct image planes as input, and produces one or more image planes as output. Input can be taken from any data planes in the training data image cube. Output is written to any of a small number of *scratch planes* — temporary workspaces where an image plane can be stored. Genes can also take input from scratch planes, but only if that scratch plane has been written to by another gene earlier in the chromosome sequence.

The image processing algorithm represented by any particular chromosome can be thought of as a directed acyclic graph, where the non-terminal nodes are primitive image processing operations, and the terminal nodes are individual image planes extracted from the multi-spectral image used as input. The scratch planes are the "glue" that combines primitive operations into image processing pipelines. Traditional GP [17] uses a variable sized (within limits) tree representation for algorithms. Our representation differs in that it allows for re-use of values computed by sub-trees, since many nodes can access the same scratch plane, i.e., the resulting algorithm is a graph rather than a tree. It also differs in that the total number of nodes is fixed.

Our notation for genes is most easily illustrated by an example: the gene [ADDP rD1 rS1 wS2] applies pixel-by-pixel addition to two input planes, read from data plane 1 and from scratch plane 1, and writes its output to scratch plane 2. Additional operator parameters, if any, are listed after the input and output arguments.

Our "gene pool" is composed of a set of primitive image processing operators which we consider useful. For different applications, the user may want to choose different sets of primitive operators; for the studies described here, we used the operators described in Table I. These include spectral, spatial, spatio-spectral, logical and thresholding operators.

The set of morphological operators is restricted to function-set processing morphological operators, i.e. grayscale morphological operators having a flat structuring element. The shape of the structuring elements used by these operators is chosen from among: square, circle, diamond, horizontal cross and diagonal cross, and horizontal, diagonal and vertical lines. The shape and size of the structuring element are defined by operator parameters. Other local neighborhood/windowing operators such as mean, median, etc. specify their kernels/windows in a similar way. The spectral operators have been chosen to permit weighted sums, differences and ratios of data and/or scratch planes.

It should be noted that although all chromosomes have the same fixed number of genes, the *effective length* of the

Gene Abbreviation	Image Processing Operation	Inputs/ Outputs/ Parameters	Notes
ADDP	Add planes	2/1/0	Basic mathematical operations. ADDS adds a scalar, which may
ADDS	Add scalar	1/1/1	be negative, to its input. DIFF is like SUBP but outputs the
SUBP	Subtract planes	2/1/0	absolute values. NDI is like SUBP, but divides the result by the
DIFF	Absolute difference	2/1/0	sum of its two inputs. MULTS scales its input by a scalar, which
NDI	Normalized difference	2/1/0	by default is positive. LINSCL is like MULTS but takes an extra
MULTS	Multiply by scalar	1/1/1	parameter which is added onto the scaled input. LINCOMB
NEG	Negate plane	1/1/0	outputs a linear combination of its two inputs, in proportion
MULTP	Multiply planes	2/1/0	specified by its one parameter, which takes a value between 0
SQRT	Square root	1/1/0	and 1.
SQR	Square	1/1/0	
LINSCL	Linear scale	1/1/2	
LINCOMB	Linear combination	2/1/1	
MIN	Minimum	2/1/0	Logical operations. MIN and MAX perform pixel-wise minimum
MAX	Maximum	2/1/0	and maximum, equivalent to AND and OR for binary input.
IFLTE	'If less than else'	4/1/0	IFLTE outputs its third input wherever the first input is less than its second input, and its fourth input elsewhere.
CLIP_HI	Clip high	1/1/1	Thresholding operations. CLIP_HI truncates any pixel values
CLIP_LO	Clip low	1/1/1	above a value set by its parameter. CLIP_LO does the converse.
THRESH	Threshold	1/1/1	THRESH sets all values below its threshold parameter to 0, and all those above to `dataScale`.
SAVAR	Spectral angle variance	2-16/1/2	Spectral angle operations. SAVAR and SADIST look at two
SADIF	Spectral angle difference	2-16/1/2	circular neighborhood regions around each pixel, of size defined
SADIST	Spectral angle distance	2-10/1/2-10	by their two parameters. SAVAR returns the difference between
SANORM	Normalize spectral vector	2-10/2-10/0	the variance of the spectral angles of the pixels in the two regions. SADIF returns the difference between the mean spectral angle of both regions. SADIST returns the spectral angle difference between each pixel and the vector defined by its parameters. SANORM normalizes the vector defined by its inputs to have a magnitude equal to `dataScale`.
QTREG	Region Size related to Statistics	1/3/1	QTREG Determines the region size (in log base 2) around each pixel for which the normalized variance per pixel standard of the square region first reaches a given threshold. Also returns planes with the linear fit slope and offset of the variance as a function of region scale for each pixel
R5R5	Laws' texture measure	1/1/0	Neighborhood operations. In general, all these operations take a
LAWB	Laws' texture measure	1/1/0	single plane as input and produce a single output plane. The
LAWD	Laws' texture measure	1/1/0	output at each pixel is determined by looking at the pixel's
LAWF	Laws' texture measure	1/1/0	neighborhood. R5R5, LAWB, LAWD, LAWF and LAWH are
LAWH	Laws' texture measure	1/1/0	widely-used texture measures, developed by Laws, that return
LAPLAC3	3x3 Laplacian	1/1/0	zero if the neighborhood contains all the same value of pixel, and
LAPLAC5	5x5 Laplacian	1/1/0	some other value otherwise, depending upon the distribution of
MORPH_LAPLAC	Morph. Laplacian	1/1/2	pixel values. R5R5 is corresponds to Laws' $R5^T \times R5$ 5×5
ISO_GRAD	Isotropic gradient	1/1/0	operator. The others are 3×3 operators, corresponding to Laws'
MEAN	Mean	1/1/1	$S3^T \times L3$, $E3^T \times E3$, $L3^T \times S3$ and $S3^T \times S3$ operators
VAR	Variance	1/1/2	respectively. For details regarding Laws' textural operators, the
SKEWNESS	Skewness	1/1/2	interested reader is referred to [18], [19]. Most of the other
KURTOSIS	Kurtosis	1/1/2	operators are familiar image processing or morphological
SKEW_COEFF	Skewness coefficient	1/1/2	operators, whose description can be found in any good book on
KURT_COEFF	Kurtosis coefficient	1/1/2	image processing. Most take two parameters which give the size
SD	Standard deviation	1/1/2	and shape of a structuring element defining the neighborhood to
RANGE	Morphological Gradient	1/1/2	which the operator is applied. ASF stands for 'Alternating
MEDIAN	Median	1/1/2	Sequential Filter'. MB_EDGE takes an additional parameter
EROD	Erode	1/1/2	defining a threshold for edge strength to be looked for. The
DIL	Dilate	1/1/2	single parameter for H_DOME and H_BASIN defines the pixel
OPEN	Open	1/1/2	value offset used by these operators.
CLOS	Close	1/1/2	
OPCL	Open-close	1/1/2	
CLOP	Close-open	1/1/2	
ASF_CLOP	ASF Close-open	1/1/2	
ASF_OPCL	ASF Open-close	1/1/2	
POS_TH	Positive top hat	1/1/2	
NEG_TH	Negative top hat	1/1/2	
OP_REC	Open with reconstruction	1/1/2	
CL_REC	Close with reconstruction	1/1/2	
H_DOME	H-dome	1/1/1	
H_BASIN	H-basin	1/1/1	
MB_EDGE	Canny edge detector	1/1/2	

TABLE I

THE PRIMITIVE IMAGE PROCESSING OPERATORS (GENES) USED IN GENIE AND WHAT THEY DO.

(a) (b)

Fig. 1. (a) Gray-scale images of one of the scenes used to produce the training data for "Urban Areas" (Urban 1) (b) Training data provided for the training scene for "Urban Areas" (White = Feature, Grey = Not Feature, Black = No Assertion)

resulting algorithm graph may be smaller than this. For example, an operator may write to a scratch plane that is then overwritten by another gene before anything has a chance to read from it. GENIE performs an analysis of chromosome graphs when they are created and only carries out those processing steps that actually affect the final result. Therefore, the fixed length of the chromosome acts as a maximum effective length.

In an interesting parallel to "junk DNA" in natural chromosomes, the final chromosomes produced by GENIE often exhibit some redundancy, i.e. genes and answer planes that do not contribute to the answer. While these "junk genes" do not affect the functionality of the chromosome, they can make it harder to understand how the chromosome works. We have therefore developed a simple post-run pruning process that removes junk genes and ineffective answer planes from the final solution if this is required.

C. Backends

Final classification requires that the algorithm produce a single scalar output plane, which can then be thresholded to produce a binary output. It would be possible to treat, for example, the contents of scratch plane **S1** as the output from the algorithm (thresholding of this plane may be required to obtain a binary result). However, we have found it advantageous to adopt a hybrid approach which applies a conventional supervised classifier to a (sub)set of scratch and data planes to produce the final output plane.

To do this, we first select a subset of the scratch and data planes to be *answer planes*. The conventional supervised classifier "backend" uses the answer planes as input and produces a final output plane; in principle, we can use any supervised classification technique as the backend but for the comparisons reported here, we used the *Fisher Linear Discriminant* [20]. This provides a linear combination of the answer planes that maximizes the mean separation between true and false pixels, normalized by the

total variance in the projection defined by the linear combination. The output of the discriminant-finding phase is a continuous-valued (gray-scale) image, which is then reduced to a binary image by finding the threshold value that maximizes the fitness as described in the following section.

D. Fitness Evaluation

The fitness of a candidate solution is given by the degree of agreement between the final binary output plane and the training data. If we denote the detection rate (fraction of "true" pixels classified correctly) as R_d and the false alarm rate (fraction of "false" pixels classified incorrectly) as R_f, then the fitness F of a candidate solution is given by

$$F = 500(R_d + (1 - R_f)). \tag{1}$$

Thus, a fitness of 1000 indicates a perfect classification result. This fitness score gives equal weighting to type I (true pixel incorrectly labelled as false) and type II (false pixel incorrectly labelled as true) errors. Note a fitness score of 500 can be trivially achieved with a classifier that identifies all pixels as true (or all pixels as false).

E. Software Implementation

The evolutionary algorithm code has been implemented in object-oriented Perl. This provides a convenient environment for the string manipulations required by the evolutionary operations and simple access to the underlying operating system (Linux). Chromosome fitness evaluation is the computationally intensive part of the evolutionary process and we currently farm this job out to a separate process running a commercial image processing engine (Interactive Data Language (IDL), by Research Systems, Inc. [21]). IDL does not provide all the image processing operators we want, so we have implemented additional operators in C that can be called from within the IDL environment.

Within IDL, individual genes correspond to single primitive image operators, which are coded as IDL procedures; a chromosome is a sequence of genes and exists as lines of IDL code in an IDL batch executable. In our present implementation, an IDL session is opened at the start of a run and communicates with the Perl code via a two-way UNIX pipe. This pipe is a low-bandwidth connection. It is only the IDL session that needs to access the input and training data (possibly hundreds of Megabytes), requiring a high-bandwidth connection. The ALADDIN training data mark-up tool was written in Java. Fig. 2 shows the software architecture of the system.
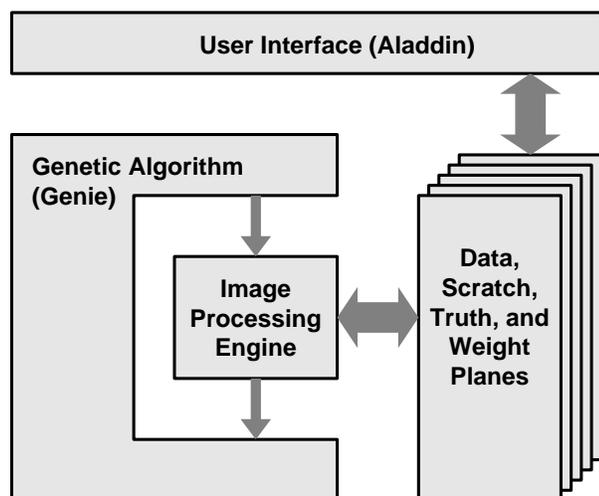
Fig. 2.   Software Architecture of the GENIE System.

III. CONVENTIONAL SUPERVISED CLASSIFICATION

Many implementations of standard supervised classifiers exist. One of the most widely used remote-sensing software packages is the ENvironment for Visualizing Imagery (ENVI) [1], which is built on IDL and is also distributed by Research Systems, Inc. Supervised classification techniques provided as part of the ENVI package were used in the comparison experiments with GENIE. Currently GENIE is set up to be trained using effectively three classes: "feature", "non-feature" and "don't care" and to be able to classify every pixel in its input data into one of two classes: "feature" and "non-feature". The normal mode of operation of the ENVI supervised classifiers is to use training data for the one "true" class: i.e. the feature of interest. The ENVI classifier is then used to classify the input image into "feature" or "unclassified". The user adjusts the parameters of the particular supervised classifier in order to attain optimal performance, with respect to feature identification. For our experiments, these parameters were adjusted to maximize the fitness defined in Eq. 1.

The one exception to this is the Maximum Likelihood classifier, which requires more than one class in the training data. In this case we used the "feature" and "non-feature" classes and the Maximum Likelihood classifier classified every pixel in the input data into one or other of these two

classes, with no "unclassified" pixels being allowed. For applying the ENVI-supplied classifiers to out-of-training-sample data, the training data (reference spectra) used in the training was provided, together with the parameters that gave optimal performance on the training data. For the GENIE case, it was simply a case of applying the algorithms found by GENIE to the out-of-training-sample data (including the linear discriminant and threshold found during training).

In Section VI, we show auxiliary results from training the ENVI classifiers with more than just these two ("feature" and "non-feature") classes.

The following ENVI-supplied supervised classification techniques were used in the comparison experiments [22].

A. (MIN) Minimum Distance

The minimum distance supervised classification technique [22], [23] computes the mean pixel vector of the "feature" class, and then assigns new pixels to the "feature" class based on the Euclidean distance from that pixel to the mean. For the multi-class case, the pixel is assigned to the feature whose mean value is the minimum distance from the pixel. For the simple feature/non-feature discrimination here, the pixels is identified as a "feature" if the distance is less than a user-defined threshold (adjusted to obtain optimum performance on the training data); otherwise, it is a "non-feature".

B. (MAX) Maximum Likelihood

Maximum likelihood classification is the most common supervised classification method used with remote sensing data [23], and among the classifiers considered here, the one with the most free parameters. Here each class ("feature" and "non-feature") is modelled with separate multivariate gaussian distributions. New pixels are assigned to the class that had the highest probability of generating that pixel.

C. (MAH) Mahalanobis Distance

The Mahalanobis distance technique [23] is very similar to the maximum likelihood classifier, but with the simplification that all classes are modelled as having identical covariance matrices (which define the shape and orientation of the normal distribution). In the one class case, we compare the probability that a new pixel was generated by the "feature" class, to a user-defined threshold, in order to decide the class to which each pixel belongs.

D. (SAM) Spectral Angle Mapper

The spectral angle mapper (SAM) technique [24] is motivated by the observation that changes in illumination caused by shadows, slope variation, sun position, light cloud, etc., approximately only alter the magnitude of a pixel's vector, rather than the direction. Therefore we can eliminate these effects by normalizing all pixel vectors to unit magnitude and then looking at the angle between a given pixel and the mean vector for the "feature" class. Pixels are assigned to the "feature" class if this angle is less than a user-defined threshold.

E. (BIN) Binary Encoding

Binary encoding classification [23], [25] encodes the data and reference spectra into ones and zeros, based on whether a particular band value lies above or below the spectrum mean. The comparison between the encoded reference spectrum with the encoded data spectra is performed using a Boolean logic exclusive OR (XOR) function. A user specifies the minimum fraction of bands that must match between the encoded reference spectrum and the data spectra. Pixels that do not meet this criterion are labeled as "non-feature". We note that binary encoding produces an extreme coarsening of the data. It was invented for, and is most appropriately applied to, hyperspectral data.

It is worth noting that for the traditional supervised classifiers, the user-defined thresholds determined as being optimal for the training data may not be optimal for out-of-training-sample data. However, we can envisage a production scenario, where the classifiers are trained on one set of data to find a particular feature, where some kind of "ground truth" is available and the resultant classifier is applied to some other out-of-training-sample data, in order to determine if that particular feature is present or not in the data, and "ground truth" data not be available for that data. In this case, the lack of ground truth means that there is no quantitative way of determining the optimal threshold value for the out-of-training-sample data. It should also be pointed out that this is also the case for the GENIE classifiers. GENIE's backend has a threshold which needs to be determined and the value determined as optimal for a training set may not be optimal for out-of-training-sample data. So, for a fair comparison, thresholds determined for all classifiers during training where left unchanged when the classifiers were applied to out-of-training-sample data. In addition, experiments were also conducted in which user-adjusted thresholds were not employed, where the traditional classifiers were forced to classify the entire scene into feature or non-feature based on the particular distance measure appropriate to the classifier. This amounts to a planar separating surface compared to a sphere for the user-defined threshold case. It was found that the user-adjusted threshold scenario performed better, in general.

IV. EXPERIMENTAL DATA AND CLASSIFICATION TASKS

A. Data Used in the Experiments

The remotely-sensed images referred to in this paper were derived from the Airborne Visible and InfraRed Imaging Spectrometer (AVIRIS) [26], a sensor developed and operated by the NASA Jet Propulsion Laboratory. The AVIRIS sensor collects data in 224 contiguous, relatively narrow (10 nm), uniformly-spaced spectral channels. AVIRIS is an airborne sensor and spatial resolution can vary from a few meters to 20 meters, depending on the altitude of the collecting platform. We used data from 1996 and 1997 AVIRIS campaigns from a range of sites shown in Table II; more detail is available from the AVIRIS quick-look website [27].

For the studies reported here, we used a reduced number of relatively wide spectral bands, designed to simulate imagery from a new remote sensing satellite called the Multispectral Thermal Imager (MTI) [28]. The MTI satellite was launched in March 2000 and collects data in 15 spectral bands. Ten of these bands sample wavelengths between 0.4 and 2.4 microns, a region covered by the AVIRIS instrument. As test data to develop analysis codes for the MTI mission, AVIRIS data were convolved with the MTI spectral filter functions to produce simulated MTI data. This 10-band simulated data was used for development of both conventional remote sensing algorithms and for GENIE development, such as reported here.

Feature	Site Name	AVIRIS Flight Number
Roads 1	Moffet Field	f970620t01p02_r03_sc01
Roads 2	ARM Site	f970801t01p02_r01_sc06
Roads 3	Denver	f970701t01p02_r07_sc07
Golf 1	Moffet Field	f970620t01p02_r03_sc01
Golf 2	Denver	f970701t01p02_r07_sc01
Golf 3	Kennedy Space Center	f960323t01p02_r04_sc02
Urban 1	Denver	f970701t01p02_r07_sc09
Urban 2	Denver	f970701t01p02_r07_sc04
Urban 3	Moffet Field	f970620t01p02_r03_sc01
Clouds 1	Atlanta	f970806t01p02_r07_sc01
Clouds 2	Atlanta	f970806t01p02_r07_sc02
Clouds 3	"Clouds 1"	f970817t01p02_r05_sc01

TABLE II
LIST OF DATA SETS USED IN THE EXPERIMENTS

The images displayed here are false-color images (which have then been converted to gray-scale in the printing process). The color mappings used are the same for all original image data shown. The particular color mappings used here involve averaging MTI bands A (0.45–0.52 μm) and B (0.52–0.60 μm) for the blue component, bands C (0.62–0.68 μm) and D (0.76–0.86 μm) for the green component and bands E (0.86–0.89 μm) and F (0.91–0.97 μm) for the red component. In addition, the images have been contrast enhanced. The choice of color mappings was arbitrary, in that it was a personal decision made by the analyst in order to best "highlight" the feature of interest, and thereby enable the production of high quality training data. This ability to manipulate the image with color mappings and contrast enhancement is an important feature of the graphical interface.

B. Classification Tasks

We chose four different features of interest: roads, golf courses, urban areas, and clouds. These features were chosen because of their particular attributes in multi-spectral data. The features were considered a good test of a supervised classification technique due to the different levels of difficulty they posed for these techniques. Clouds are relatively easy, and mostly spectral; urban areas encom-

pass a land-cover distinction; roads are easy for the eye to find, but notoriously difficult for automated algorithms; golf courses require a combination of spectral and spatial information to disambiguate them from other similarly-vegetated areas (e.g. lawns).

We set the various supervised classification techniques the task of distinguishing these features within several scenes of the 10-channel multi-spectral data as described above. For each feature of interest three separate scenes had training data marked-up using the ALADDIN tool. This provided "ground truth" for training data and for assessing the performance of the classification scheme on out-of-training-sample data. We employed a cross-validation scheme where, for each feature, we trained a classifier separately on the three marked-up scenes, and then for each scene, applied the resulting classifier to the two remaining out-of-sample scenes. GENIE was run, with a population of 100 individuals, for 500 generations, or until a (perfect score) fitness of 1000 was achieved.

An example of an image plus associated training data is shown in Fig. 1. This figure shows the false-color image for one of the scenes used for the "urban area" feature classification, and the associated training data. In the training data image the white pixels correspond to the places on the image where the feature is asserted to be, the grey pixels to where the feature is asserted not to be, and the black pixels correspond to places where no assertion is made.

V. COMPARISON EXPERIMENTS

For the training phase, we ran GENIE and the ENVI-supplied classifiers on the training data. For GENIE, the result of this training phase is an image processing pipeline which can be applied to and tested on other data. To apply the ENVI-supplied classifiers to out-of-training-sample data it was necessary to save the regions of interest of the marked-up training classes and provide them as the reference spectra for application of the classifiers to out-of-training-sample data.

We measured the fitness, detection rate and false-alarm rate of all the classifiers on the training data and out-of-training-sample data. Table III summarizes the quantitative results of the comparison between the GENIE algorithm output and the traditional algorithms' output for each of the features. The bottom four rows of the table show the average, for each classification technique, across all features sought. It is interesting to notice that the relative ranking (based on fitness score) of each of the classifiers is relatively stable over the different features, with the more complicated classifiers generally achieving the highest scores. For the out-of-training-sample data, by contrast, the simpler algorithms (with fewer free parameters) perform much better. The main exception is GENIE, which performs well on both the traning data and on the out-of-training-sample data.

An example of an image processing pipeline produced by GENIE is given by the following solution to the golf course-finding task:

[QTREG rD7 wS5 wS3 wS1 0.05]*

[MEAN rD4 wS2 4 0]
[MEAN rS2 wS2 3 0]
[VAR rD7 wS4 3 0]
[CLOP rS2 wS2 3 0]
[RANGE rD10 wS1 3 0]
[OP_REC rS4 wS4 3 0]
[ASF_OPCL rD2 wS3 3 0]*

As described in Section II-B, each line consists of a single primitive image processing operation: the name of the operator, which data (D) or scratch (S) planes were read (r) from and which were written (w) to, and what parameter values were used (see Table I for details on the individual operators). GENIE produced a solution with five answer planes, and the backend produced a linear combination of those planes, along with a threshold value, to give a binary classification. A graphical representation of this pipeline is illustrated in Fig. 3. Note that the circled **D**s represent the input data planes and the circled **S**s represent the answer planes that are input to the back-end classifier (Fisher Linear Discriminant plus threshold), to produce the final classification result. To aid clarity, we now provide a narrative description of the operation of this pipeline.

The RANGE operator computes the difference between the maximum and minimum value in a 7×7 kernel of data plane D10, and writes the result to scratch plane S1. The parameters "3 0" correspond to a square 7×7 kernel. The first integer parameter for this operator, "3", actually defines the "radius" of the smoothing kernel, where the "diameter" of the kernel is always an odd integer, and defined as $(2 \times \text{radius}) + 1$. The second integer parameter, "0", defines the particular choice of kernel shape, in this case a square. A "1" would define a circle, "2" a vertical cross, "3" a diagonal cross, etc..

The first MEAN operator, [MEAN rD4 wS2 4 0], smooths the data plane D4 with a 9×9 square kernel, and writes the solution to scratch plane S2. The second MEAN operator, [MEAN rS2 wS2 3 0], smooths the result stored in the S2 plane with a 7×7 square kernel, and the CLOP operator performs a morphological close-open operation, again with a 7×7 square kernel, writing the output to scratch plane S2.

The ASF_OPCL operator performs an alternating sequential open-closing with a square kernel of maximum size 7×7 on data planeD2, and writes the output to scratch plane S3.

The VAR operator computes the variance in a 7×7 kernel of data plane D7, and writes the result to scratch plane S4. That plane is further modified by the OP_REC operator, which performs a morphological opening with reconstruction, again based on a 7×7 kernel.

The QTREG operator also reads data plane D7 and writes three scratch planes (S5, S3, and S1), two of which (S3 and S1) are overwritten by other operators before being used.

Finally, the Fisher Discriminant backend applies a linear combination of the scratch planes, followed by a threshold, to produce a binary answer plane. The coefficients applied to the five answer planes (S0, S1,S2, S3, S4) are: $\{-9.354 \times 10^{-6}, 1.235 \times 10^{-5}, 1.659 \times 10^{-6}, 1.460 \times$

		Training Data						Out-of-Training-Sample Data					
		GENIE	MIN	MAH	MAX	SAM	BIN	GENIE	MIN	MAH	MAX	SAM	BIN
Roads	Fitness	963.3	781.2	865.9	921.1	803.9	677.5	763.2	559.9	500.0	611.2	566.0	587.1
	DR (%)	96.61	78.63	83.11	91.62	82.93	82.00	60.71	27.54	0.00	62.25	30.29	72.94
	FAR (%)	3.95	22.40	10.03	7.40	22.14	46.50	7.36	15.57	0.00	40.02	17.09	55.51
	Rank	1st	5th	3rd	2nd	4th	6th	1st	5th	6th	2nd	4th	3rd
Golf	Fitness	998.3	945.0	947.6	966.2	915.4	820.1	739.8	584.8	500.0	553.1	696.5	572.9
	DR (%)	99.68	95.16	93.84	96.21	92.73	78.72	61.60	42.34	0.00	10.93	58.76	51.16
	FAR (%)	0.00	6.17	4.32	2.97	9.64	14.71	13.65	25.39	0.00	0.31	19.46	36.57
	Rank	1st	4th	3rd	2nd	5th	6th	1st	3rd	6th	5th	2nd	4th
Urban	Fitness	998.9	694.7	861.6	963.6	636.3	580.4	813.5	586.2	514.6	569.4	499.1	521.9
	DR (%)	99.85	58.55	80.34	95.67	75.03	83.59	66.32	27.36	2.93	65.86	50.67	70.18
	FAR (%)	0.07	19.61	8.03	2.94	47.77	67.52	3.63	10.11	0.02	51.97	50.51	65.80
	Rank	1st	4th	3rd	2nd	5th	6th	1st	2nd	5th	3rd	6th	4th
Clouds	Fitness	999.9	975.7	946.7	997.9	979.4	760.2	978.0	968.6	632.0	701.7	975.3	727.1
	DR (%)	99.99	96.41	94.21	99.91	98.18	55.59	97.43	95.38	28.62	99.97	97.28	48.85
	FAR (%)	0.00	1.27	4.86	0.33	2.29	3.55	1.82	1.67	2.22	59.64	2.23	3.43
	Rank	1st	4th	5th	2nd	3rd	6th	1st	3rd	6th	5th	2nd	4th
Average	Fitness	990.1	849.1	905.5	962.2	833.8	709.5	823.6	674.9	536.6	608.8	684.2	602.3
	DR (%)	99.03	82.19	87.87	95.86	87.22	74.98	71.51	48.16	7.89	59.75	59.25	60.78
	FAR (%)	1.01	12.36	6.81	3.41	20.46	33.07	6.62	13.18	0.56	37.99	22.32	40.33
	Rank	1st	4th	3rd	2nd	5th	6th	1st	3rd	6th	4th	2nd	5th

TABLE III

COMPARISON OF GENIE'S EVOLVED ALGORITHM WITH ENVI ALGORITHMS (DR = DETECTION RATE, FAR = FALSE ALARM RATE)

$10^{-9}, 0.0349\}$. There is an additional DC offset value of -0.350 applied to the output of the linear combination. The threshold value for determining the binary output was 0.664305.

It can be seen that this image processing pipeline has only used 4 of the available 10 data planes as input: data planes D2, D4, D7 and D10. These correspond to the MTI bands B (0.52–0.60 μm), D (0.76–0.86 μm), G (0.99–1.04 μm) and O (2.08–2.35 μm) respectively. GENIE's choice of input data bands is (in retrospect) not too surprising, given the task. The algorithm is using the green band (B), as well as two near-infrared (NIR) bands (D,G) and a short-wave infra-red (SWIR) band (O). Vegetation is highlighted in the two NIR bands that GENIE selected, as well as in the green band.

Of these five answer planes the most important were S1, S2 and S4; using only those planes we could still achieve the same fitness value, on the training data and out-of-training-sample data, as when all the answer planes were used. Hence, two of the operators (those marked above with asterisks) did not contribute substantially to the solution. The outputs of the useful answer planes, as can be seen from Fig. 3, are derived from the NIR and SWIR bands. In this case we see, somewhat surprisingly, that the green band is not contributing significantly to the solution. We might expect green to be very useful for identifying golf courses, and this is probably how it made its way into the chromosome. However, in the end, the NIR and SWIR bands were found to be more informative.
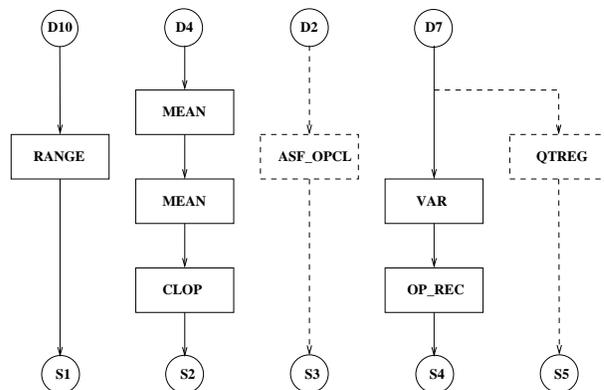
Fig. 3. Image processing pipeline discovered by GENIE for finding golf courses. Dotted lines indicate scratch planes which did not contribute significantly to the final classification.

We illustrate the results of these classification techniques on training and out-of-training-sample data with an example of output from GENIE, and from the best-performing ENVI classifier, on the golf course problem. Figs. 4 and 5 compare GENIE to Maximum Likelihood (MAX) for one of the training data sets, and Figs. 6 and 7 compare GENIE to the Spectral Angle Mapper (SAM) on out-of-training-sample data.

An interesting aspect of GENIE's performance to consider is its repeatability: i.e. whether or not, for a given feature, GENIE leads to the same result (i.e., the same "image processing pipeline") when trained on different scenes. In
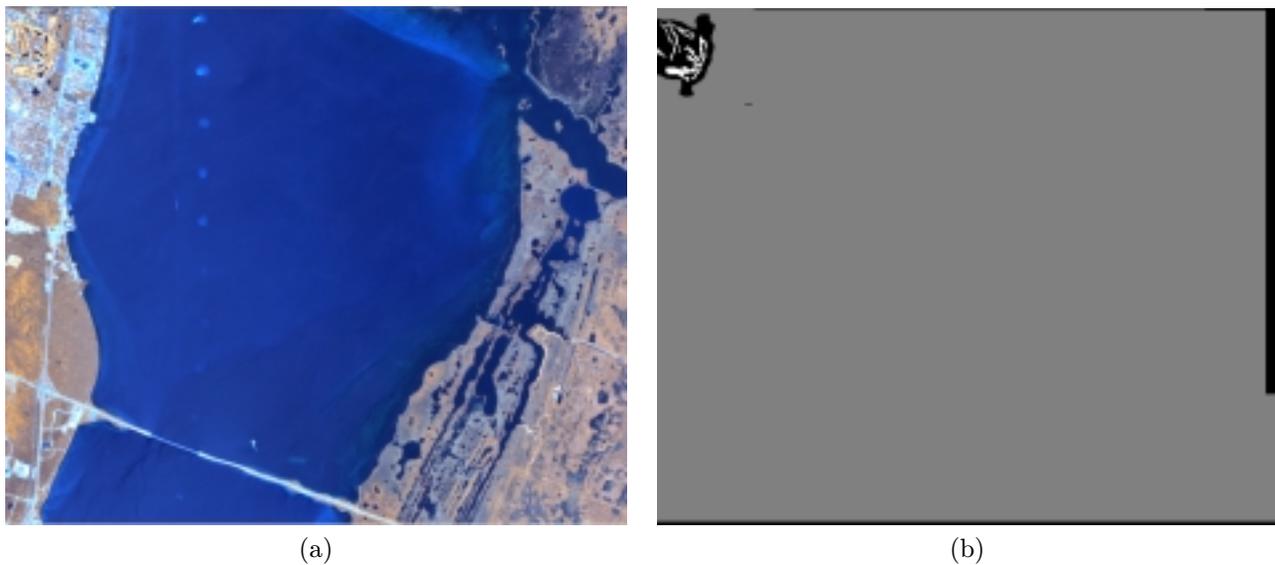
Fig. 4. (a) Gray-scale images of one of the scenes used to produce the training data for "Golf Courses" (Golf 3) (b) Training data provided for the training scene for "Golf Courses" (White = Feature, Grey = Not Feature, Black = No Assertion). The black "buffer area" around the golf course reflects the analyst's lack of concern with a detailed delineation of the precise extent of the golf course.
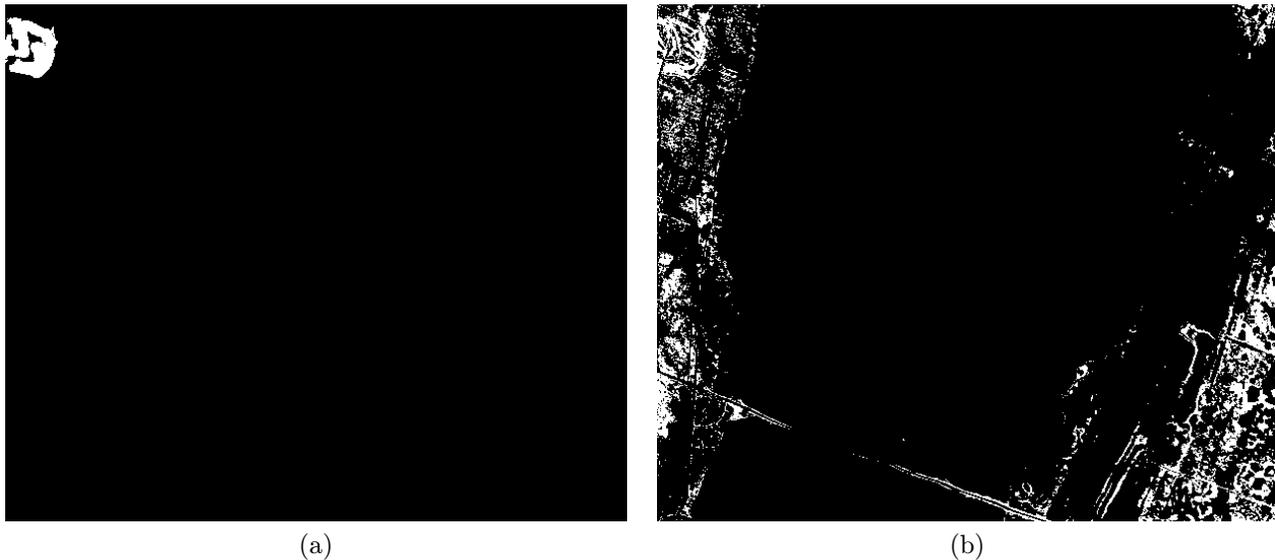
Fig. 5. (a) GENIE results on training data: Fitness = 999.2 (b) Best ENVI classifier for the particular training scene (Minimum Distance): Fitness = 957.4. Here GENIE's use of spatial information is clearly evident. The ENVI classifier actually did a better job of delineating the extent of the golf course, whereas GENIE's spatial operators led to a "fatter" golf course than the purely spectral data would warrant. On the other hand, though, this spatial information allowed GENIE to veto the golf course-like spectra in the rest of the image. Because the "fatter" golf course fits inside the no-assertion region, GENIE is not penalized.

general, GENIE will not produce the same image processing pipeline even when trained on the same scene, if it starts with a different random number seed. However, the different solutions will generally have the same approximate performance, both on training data and on out-of-sample data, and there will often be an overlap in the choice of operators and data planes used in the image processing pipeline that is evolved. But the space of image processing pipelines it too large and too rugged to achieve any real level of "robustness", in this regard.

VI. Further Experiments and Results

Depending on the application at hand, an image analyst is sometimes interested in the identification of a single specific feature against a background of everything else in the image, and is sometimes interested in the simultaneous extraction of multiple features (for instance, when making a landcover map). The experiments described in the previous sections take the first point of view and it is this binary classification task that GENIE was designed to handle.

However, for Maximum Likelihood and other conventional classifiers, the "background of everything else" is not well modelled as a single unimodal class. To address
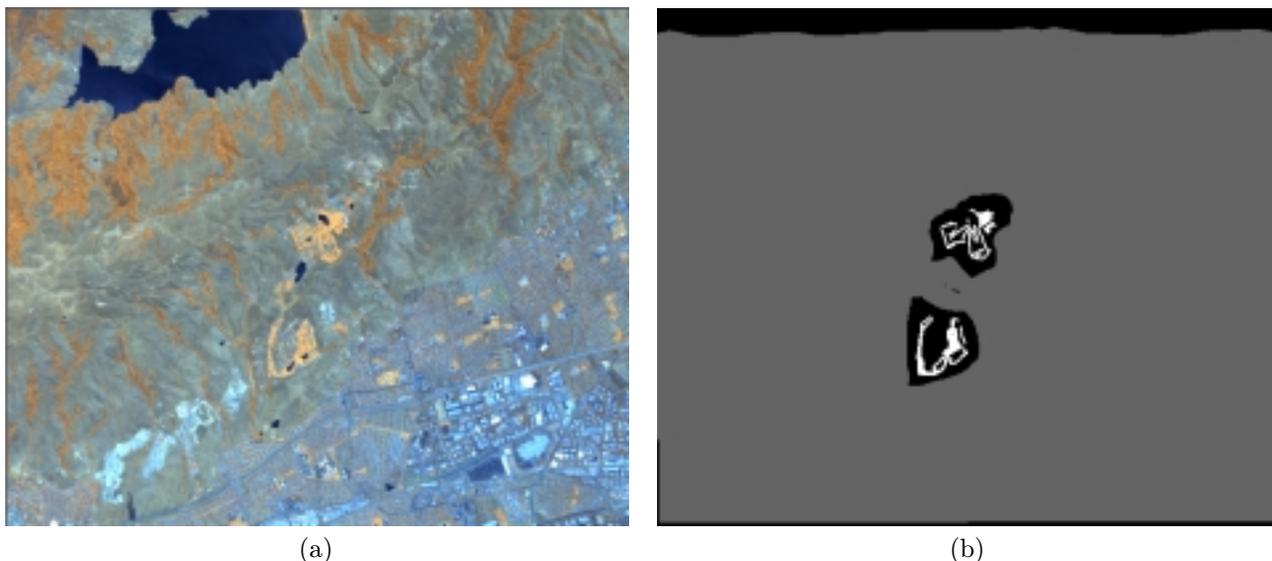
Fig. 6. (a) Gray-scale images of one of the scenes used to produce training data for "Golf Courses" (Golf 1) (b) Training data provided for the training scene for "Golf Courses" (White = Feature, Grey = Not Feature, Black = No Assertion)
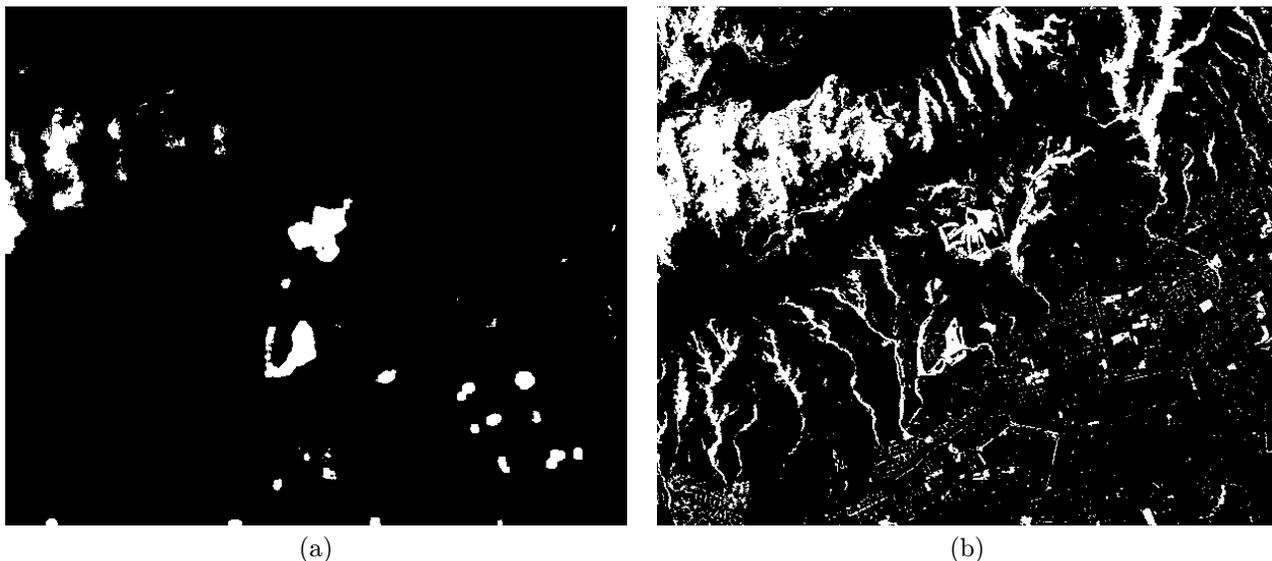
Fig. 7. (a) GENIE results on out-of-training-sample data: Fitness = 946.9 (b) Best ENVI classifier (for particular training scene) on out-of-training-sample data (Spectral Angle Mapper): Fitness = 856.7. Again, Genie has used its spatial operators to produce "fatter" golf courses, but it was also able to censor more of the non-golf-course area in the rest of the scene.

this difficulty, it has been suggested [29] to artificially divide the background into multiple classes, and then employ multi-class classification techniques. This combined use of labelled and unlabelled samples can often lead to more powerful supervised classification [30], [31], [32], [33].

In order to address these same issues, we conducted a series of further experiments where we adopted a similar approach, in which the standard supervised classification techniques were given the task of classifiying the scenes into multiple classes instead of the two feature/non-feature classes described in the previous experiments.

A. Experimental Procedure

The training data as provided to GENIE and as used in the experiments described in Section V were used to cre-ate the training data provided to the standard supervised classifiers. The "feature" class was kept as it was, but the "non-feature" class was divided up into multiple classes. The combination of the "feature" class and the sub-divided "non-feature" class was then given as training data to the standard supervised classification techniques.

It should be noted that the binary encoding supervised classification technique was not included in these additional experiments.

The "non-feature" class was divided into multiple classes by applying ENVI's unsupervised k-means classification algorithms [1], [34], [35] to the entire "non-feature" class. This k-means classification was performed several times, varying the number of classes into which the non-feature class was classified. The k-means classification with the

number of classes that provided the best final classification performance in terms of fitness, was the one included in the additional results shown here.

Table IV shows the overall ranking for the multiple-class classification algorithms, averaged over all the features for the training data. In this table, "(M)" indicates the use of multi-class training; the non-multiple-class results are those results described and shown in Section V.

	Training		Out-of-Training-Sample	
Rank	Classifier	Fitness	Classifier	Fitness
1st	GENIE	990.1	GENIE	824.5
2nd	MAX (M)	965.8	SAM (M)	780.0
3rd	MAH (M)	949.1	MIN (M)	778.7
4th	MAH	905.5	MAH (M)	752.4
5th	MIN (M)	892.3	SAM	684.2
6th	SAM (M)	874.1	MIN	674.9
7th	MIN	849.1	MAX (M)	615.5
8th	SAM	833.8	MAH (M)	536.6

TABLE IV

RANKINGS, BASED ON FITNESS SCORES AVERAGED OVER ALL CLASSIFICATION TASKS, OF THE BEST MULTIPLE-CLASS VARIANTS OF THE STANDARD SUPERVISED CLASSIFIERS; ON BOTH TRAINING DATA AND TESTING DATA

VII. DISCUSSION

With a single exception, GENIE out-performed all the other classification techniques on both training data and out-of-training-sample data, for all of the classification tasks considered. For the training data, the gap, with respect to fitness, between GENIE's performance and the best of the other techniques was much less than for the out-of-training-sample case. This suggests that GENIE is significantly better at generalizing than the other techniques compared here. An interesting observation is that the best of the other techniques on the training data did not necessarily guarantee it to be the best of the other techniques on the out-of-training-sample data. This indicates the sensitivity of these techniques to training data and highlights GENIE's generalization abilities.

The one exception was the multi-class Spectral Angle Mapper applied to golf courses, on out-of-training-sample data. This suggests that golf courses are relatively well identified by their spectral signatures (perhaps not surprising in a desert/mountain environment where they are quite distinctive), and that the illumination-invariance built into the Spectral Angle Mapper provided it the edge to better generalize to other scenes. Since GENIE was trained on only one scene at a time, it did not "learn" to employ an illumination-independent solution.

One issue to be addressed is training time. At present GENIE requires the testing of potentially thousands of candidate algorithms on the training data. Depending on the size of the data, this can take hours to complete. This is considerably longer to train than the other techniques. It

should be noted, though, that the result of GENIE's training is an image processing algorithm that can be applied to other data with times comparable to that of the other techniques' application to out-of-training-sample data. We also remark that a few hours is usually a small fraction of the time it would take to hand-design an equivalent image processing pipeline that is customized not only to the specific feature, but also to the specific data set. Another point to consider is that being a population-based optimization technique, GENIE lends itself well to parallelization, which can dramatically reduce training time. Some experiments have been carried out to demonstrate this [6].

Although the traditional classification techniques that were compared here use only spectral information, it is possible to enable these techniques to use spatial information as well. There is in fact a large literature on methodologies for combining spatial and spectral information (eg, see Refs. [36], [37], [38], [39]). Our approach was to apply a set of spatial operators to each plane in the input multispectral data and then combine these new processed data planes with the raw data planes; both sets of planes would then be provided as input to the supervised classifiers. We applied a number of morphological smoothings at different scales to the input data and combined this with the original data. We found that this information did improve the fitness scores achieved by the conventional supervised classifiers, but they were still considerably below the performance of GENIE on the original data. Also, the improved performance was only for the training data. The classifiers actually performed worse on out-of-training-sample data (i.e., they were less robust). Obviously, if one were to adopt this approach, the choice of which spatial operators to apply is very important and the search space in this regard is immense. If one considers a scenario where some sophisticated technique is used to search the space for the optimal combination of spatial operators, one is entering the arena in which GENIE is designed to function.

In conclusion, an automated feature detection/classification system based on genetic programming has been described. Experiments comparing this new system with traditional supervised classifiers indicate consistently better performance, on both training data and out-of-training-sample data. We attribute GENIE's success to the choice of solution representation – as a multispectral image processing pipeline – and to the fact that it very naturally combines information from both the spectral and spatial domains.

REFERENCES

[1] http://www.rsinc.com/envi/index.cfm
[2] S.P. Brumby, J. Theiler, S.J. Perkins, N.R. Harvey, J.J. Szymanski, J.J. Bloch and M. Mitchell, "Investigation of Image Feature Extraction by a Genetic Algorithm," *Proc. SPIE*, vol 3812, pp. 24–31, 1999.
[3] J. Theiler, N.R. Harvey, S.P. Brumby, J.J. Szymanski, S. Alferink, S. Perkins, R. Porter and J.J. Bloch, "Evolving Retrieval Algorithms with a Genetic Programming Scheme," *Proc. SPIE*, vol 3753, pp. 416–425, 1999.
[4] N.R. Harvey, S. Perkins, S.P. Brumby, J. Theiler, R.B. Porter, A.C. Young, A.K. Varghese, J.J. Szymanski and J.J. Bloch, "Finding Golf Courses: The Ultra High Tech Approach," *Lecture Notes in Computer Science*, vol. 1803, pp. 54–64, 2000.

[5] S.P. Brumby, J. Theiler, N.R. Harvey, S.J. Perkins, R.B. Porter, J.J. Szymanski and J.J. Bloch, "A Genetic Algorithm for Combining New and Existing Tools for Multispectral Imagery," *Proc. SPIE*, vol 4049, 2000.

[6] N.R. Harvey, S.P. Brumby, S.J. Perkins, R.B. Porter, J. Theiler, A.C. Young, J.J. Szymanski and J.J. Bloch, "Parallel Evolution of Image Processing Tools for Multispectral Imagery," *Proc. SPIE*, vol 4132, pp. 72–82, 2000.

[7] K.L. Hirsch, S.P. Brumby, N.R. Harvey and A.B. Davis, "The MTI Dense-Cloud Mask Algorithm Compared to a Cloud Mask Evolved by a Genetic Algorithm and to the MODIS Cloud Mask," *Proc. SPIE*, vol 4132, 2000.

[8] S.J. Perkins, J. Theiler, S.P. Brumby, N.R. Harvey, R.B. Porter, J.J. Szymanski and J.J. Bloch, "GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multispectral Images," *Proc. SPIE*, vol 4120, pp. 52–62, 2000.

[9] C. Harris and B. Buxton, "Evolving edge detectors," *Research Note RN/96/3, University College London, Dept. of Computer Science, London*, 1996.

[10] N.R. Harvey and S. Marshall, "GA Optimization of Multidimensional Grey-Scale Soft Morphological Filters with Applications in Archive Film Restoration," in *Mathematical Morphology and its Applications to Image and Signal Processing V (ISMM2000)*, J. Goutsias, L. Vincent, D.S. Bloomberg (Eds.), pp. 129–138, Norwell, MA: Kluwer Academic Publishers, 2000.

[11] A. Teller and M. Veloso, "A controlled experiment: Evolution for learning difficult image classification," *Lecture Notes in Computer Science*, vol. 990, pp. 165-176, 1995.

[12] R. Poli and S. Cagoni, "Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement," in *Genetic Programming 1997: Proc. 2nd Annual Conference on Genetic Programming* Koza, J.R., et al. (Eds.), San Francisco: Morgan Kaufmann, 1997.

[13] J.M. Daida, J.D. Hommes, T.F. Bersano-Begey, S.J. Ross and J.F. Vesecky, "Algorithm discovery using the genetic programming paradigm: Extracting low-contrast curvilinear features from SAR images of arctic ice," in *Advances in Genetic Programming 2, chap. 21*, P.J. Angeline, K.E. Kinnear, (Eds.), Cambridge, MA: MIT Press, 1996.

[14] S. Bandyopadhyay and S. K. Pal, "Pixel classification using variable string genetic algorithms with chromosome differentiation," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, pp. 303–308, 2001.

[15] L.A. Cox Jr., L. Davis and Y. Qiu, "Dynamic anticipatory routing in circuit-switched telecommunications networks," in *Handbook of Genetic Algorithms*, L. Davis (Ed.), pp. 124–143, New York: Van Nostrand Reinhold, 1991.

[16] T. Dandekar and P. Argos, "Potential of genetic algorithms in protein folding and protein engineering simulations," *Protein Engineering*, vol. 5 pp. 637–645, 1992.

[17] J.R. Koza, *Genetic Programming : On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)* Cambridge, MA: MIT Press, 1992.

[18] K.I. Laws, "Rapid texture identification," *Proc. SPIE*, vol. 238, pp. 376-380, 1980.

[19] M. Pietikainen, A. Rosenfeld and L.S. Davis, "Experiments with Texture Classification using Averages of Local Pattern Matches," *IEEE Trans. Systems, Man and Cybernetics*, vol. SMC-13, pp. 421–426. 1983.

[20] C.M. Bishop, *Neural Networks for Pattern Recognition*, pp. 105–112. Oxford University Press, 1995.

[21] http://www.rsinc.com

[22] http://www.rsinc.com/Envi/tut2.cfm

[23] J.A. Richards and X. Jia, "Remote Sensing Digital Image Analysis," New York: Springer-Verlag, 1999.

[24] F.A. Kruse, A.B. Lefkoff, J.B. Boardman, K.B. Heidebrecht, A.T. Shapiro, P.J. Barloon, and A.F.H. Goetz, "The Spectral Image Processing System (SIPS) - Interactive Visualization and Analysis of Imaging Spectrometer Data," *Remote Sens. Environ.*, vol. 44, pp. 145–163, 1993.

[25] A.S. Mazer, M. Martin, M. Lee, J.E. Solomon, "Image Processing Software for Imaging Spectrometry Analysis," *Remote Sens. Environ.*, vol. 24, pp. 201–210, 1988.

[26] http://aviris.jpl.nasa.gov

[27] http://aviris.jpl.nasa.gov/html/aviris.quicklooks.html

[28] P.G. Weber, B.C. Brock, A.J. Garrett, B.W. Smith, C.C. Borel, W.B. Clodius, S.C. Bender, R.R. Kay, and M.L. Decker, "Mul-tispectral Thermal Imager mission overview," *Proc. SPIE*, vol 3750, pp. 340–346, 1999.

[29] B. Jeon and D.A. Landgrebe, "Partially Supervised Classification Using Weighted Unsupervised Clustering," *IEEE Trans. Geosci. Remote Sensing*, vol. 37, pp. 1073–1079, 1999.

[30] B. Shahshahani and D. Landgrebe, "Classification of Multi-Spectral Data by Joint Supervised-Unsupervised Learning," Technical Report TR-EE 94-1, School of Electrical Engineering, Purdue University, 1994.

[31] V. Castelli and T. M. Cover, "On the exponential value of labeled samples," *Pattern Recognition Lett.*, vol. 16, pp. 105–111, 1995.

[32] P.-F. Hsieh and D. Landgrebe, "Statistics enhancement in hyperspectral data analysis using spectral-spatial labeling, the EM algorithm, and the leave-one-out covariance estimator," *Proc. SPIE*, vol. 3438, pp. 183-190, 1999.

[33] L. Bruzzone and D. F. Prieto, "Unsupervised retraining of a maximum likelihood classifier for the analysis of multitemporal remote sensing images," *IEEE Trans. Geosci. Remote Sensing*, vol. 39, pp. 456–460, 2001.

[34] J.T. Tou and R.C. Gonzales, *Pattern Recognition Principles*, Addison Wesley, 1974.

[35] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, pp. 264–323, 1999.

[36] D. A. Landgrebe, "The development of a spectral-spatial classifier for Earth observational data," *Pattern Recognition*, vol. 12, pp. 165–175, 1980.

[37] B. Jeon and D. A. Landgrebe, "Classification with Spatio-Temporal Interpixel Class Dependency Contexts," *IEEE Trans. Geosci. Remote Sensing*, vol. 36, pp. 182–191, 1992.

[38] T. Yamazaki and D. Gingras, "Image Classification Using Spectral and Spatial Information Based on MRF Models," *IEEE Trans. Image Processing*, vol. 4, pp. 1333–1339, 1995.

[39] J. Theiler and G. Gisler, "A contiguity-enhanced k-means clustering algorithm for unsupervised multispectral image segmentation," *Proc. SPIE*, vol. 3159, pp. 108–118, 1997.

Neal R. Harvey received a B.Eng. (Hons.) degree in Mechanical Engineering from the University of Hertfordshire, UK in 1989, a M.Sc. in Information Technology Systems from the University of Strathclyde, UK in 1992 and a Ph.D. in Non-linear Digital Image Processing from the University of Strathclyde in 1997. He spent over 6 years as a Research Fellow in the Signal Processing Division of the University of Strathclyde's Dept. of Electronic and Electrical Engineering. In April 1999, he took up a post-doctoral research associate position in the Space and Remote Sensing Sciences Group of the Nonproliferation and International Security Division at the Los Alamos National Laboratory, NM, USA, where he is currently employed. His research interests include nonlinear digital filters, optimization techniques, machine learning, image classification, remote sensing and film and video restoration.