Hyperspectral Image Analysis Using Genetic Programming

Brian J. Ross and Anthony G. Gualtieri
Brock University
Dept. of Computer Science
St.Catharines, ON, Canada L2S 3A1
bross@cosc.brocku.ca
(905)688-5550 ext. 4284

Frank Fueten
Brock University
Dept. of Earth Sciences
St.Catharines, ON, Canada L2S 3A1
ffueten@craton.geol.brocku.ca
(905)688-5550 ext. 3856

Paul Budkewitsch
Canada Centre for Remote Sensing
588 Booth Street
Ottawa, ON, Canada K1A 0Y7
paul.budkewitsch@ccrs.nrcan.gc.ca
(613)947-1331

Abstract

Genetic programming is used to evolve mineral identification functions for hyperspectral images.[1] The input image set comprises 168 images from different wavelengths ranging from 428 nm (visible blue) to 2507 nm (invisible shortwave in the infrared), taken over Cuprite, Nevada, with the AVIRIS hyperspectral sensor. A composite mineral image indicating the overall reflectance percentage of three minerals (alunite, kaolnite, buddingtonite) is used as a reference or "solution" image. The training set is manually selected from this composite image. The task of the GP system is to evolve mineral identifiers, where each identifier is trained to identify one of the three mineral specimens. A number of different GP experiments were undertaken, which parameterized features such as thresholded mineral reflectance intensity and target GP language. The results are promising, especially for minerals with higher reflectance thresholds (more intense concentrations).

1 INTRODUCTION

Remote sensing using aircraft and satellite photography is well-established technology. The use of hyperspectral imagery, however, is relatively new. Hyperspectral images are capable of precisely capturing narrow bands of spectra through a wide range of wavelengths. Since many organic and inorganic materials exhibit unique absorption and reflection characteristics at particular bandwidths, these spectra are useful for remotely identifying various materials and phenomena of interest. This is an important area of work, since hyperspectral data permits the discovery of valuable natural resources in areas largely inaccessible by foot. Literally any area of the Earth can be mapped by hyperspectral imagery, be it with aircraft or satellites.

One complication in using this technology is the time and expertise required to interpret the data. Hyperspectral imaging systems such as the NASA/JPL AVIRIS[2] sensor can capture over 200 bandwidths for a single geographic location (Green *et al.* 1998). This is denoted by a hyperspectral cube, which takes the form of many hundreds of mega-bytes of information. Interpreting this massive amount of data is difficult, especially considering that the spectra obtained represent mixed spectral signatures of a variety of materials. Moreover, noise and other unwanted effects must be considered. Deciphering this enormous volume of cryptic data is therefore next to impossible for humans to do manually.

This paper uses genetic programming (GP) to evolve mineral classifiers for use on hyperspectral images. Separate mineral classifiers are evolved for three specific minerals – buddingtonite, alunite, and kaolinite. The classifiers take the form of programs which, when given a vector data from a particular pixel location on a hyperspectral cube, determine whether the mineral of interest resides there or not. Evolution proceeds by evaluating the performance of classifiers on positive and negative training sets. In addition, given the effects of noise at low reflectance levels, separate threshold stages are examined. This is done in the hopes that more accurate classification arises at higher reflectance levels, where there are more intense mineral concentrations.

Section 2 reviews concepts in hyperspectral imaging. The experimental design is outlined in Section 3. Section 4 presents the results of the experiments. A discussion and comparison to related work concludes the

---

[1] *Applied Soft Computing*, in press, 2004.

[2] Airborne Visible/Infrared Imaging Spectrometer.
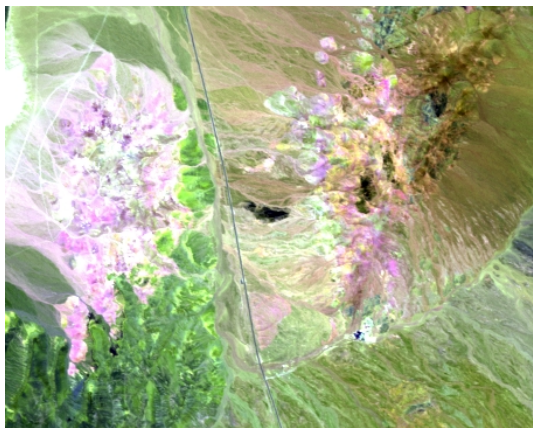
paper in Section 5.

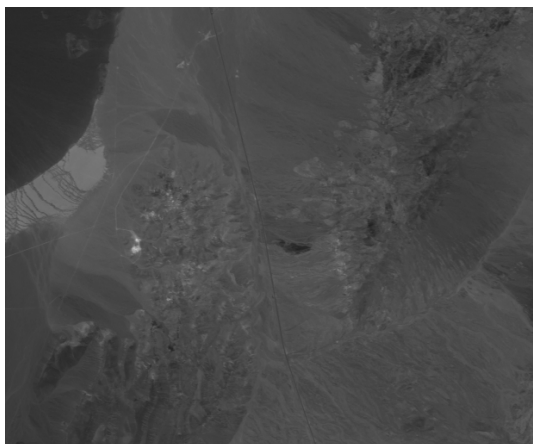## 2   BACKGROUND

Figure 1: Cuprite, Nevada.

Figure 2: AVIRIS image, $2229nm$.

The AVIRIS data used in this study was taken over Cuprite, Nevada on June 12, 1996 (19:31UT). The sensor acquires data in the wavelength region from 0.38 to 2.50 microns, with a ground sampling interval of 16.2m across track (horizontal) and 18.1m along track (vertical). At-sensor radiance data were converted to surface reflectance via an atmospheric correction using the MODTRAN3 radiative transfer (RT) code, as implemented in the imaging spectrometer data analysis system (ISDAS) (Staenz and Williams 1997). This removes spectral artifacts from solar flux and the earth's absorption bands (for example, water). This leaves surface reflectance, which is the data of interest, as it contains the spectral information pertinent to the
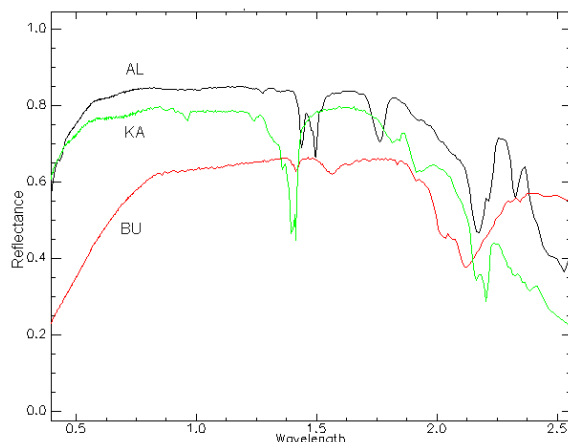

Figure 3: Spectra for alunite (AL), kaolinite (KA), and buddingtonite (BU)

identification and mapping of specific minerals and vegetation.

Figure 1 shows the Cuprite, Nevada, region studied in this paper[3]. Cuprite is a well-studied test area for remote sensing (Resmini *et al.* 1997). Figure 2 shows an AVIRIS hyperspectral reflectance image of the same area at the $2229nm$ bandwidth.

There is much ongoing research regarding the interpretation of hyperspectral reflectance imagery, and a survey is beyond the scope of this paper. A representative approach to the interpretation of reflectance data is the Tetracorder system(Clark and Swayze 1995). Identification is performed by the application of a least-squares fitting procedure to the total set of spectral data and reference spectra. Features in the absorption patterns of materials are enhanced during this fitting process, in order to promote effective identification. Multiple materials can be fitted simultaneously using this technique. Artificial neural networks are also commonly applied to the automatic analysis and classification of remote sensing data, including AVIRIS data (Ridd *et al.* 1992, Civco 1993, Dreyer 1993, Merenyi *et al.* 1993, Foody and Arora 1997, Yang *et al.* 1999, Aguilar *et al.* 2000).

Evolutionary computation has been applied to multi-spectral image analysis and remote sensing. (Larch 1994) uses genetic algorithms to evolve categorization production rules for Landsat images. (Daida *et al.* 1996) evolve genetic programs that identify ice-flow ridges from ERS SAR images. Images from aircraft are analyzed using GP in (Howard and Roberts 1999).

---

[3]North is downwards in this and all the maps in this paper.

(Rauss *et al.* 2000) evolve genetic programs for categorizing hyperspectral imagery. The GENIE system is used for hyperspectral image classification, which uses a hybrid combination of linear genetic programming with conventional classifier algorithms (Harvey *et al.* 2000, Perkins *et al.* 2000, Brumby *et al.* 2001). Among other things, GENIE has been used to classify features such as forest fires and golf courses.

Figure 3 shows the signature spectra for the minerals studied in this paper. Such spectra are measured in the laboratory, and represent reflectance signatures for various organic and inorganic materials. When minerals like these are resident in the environment, hyperspectral imaging will capture similar spectra. Tetracorder uses these lab spectra as guides for identifying minerals in hyperspectral data.

3 EXPERIMENT DESIGN

3.1 Hyperspectral data preparation

The reflectance data from Cuprite derived from the AVIRIS hyperspectral data set was analysed by (Neville *et al.* 1998). The mineral fraction maps which resulted from their work are used as the training solution for this study. From the full AVIRIS bandset available, we started with data at 0.428 microns and eliminated bands near 1.4 and 1.9, where strong absorption in the atmosphere occurs due to water vapour. This left 168 bands of data as input for our GP experiment.

3.2 Training set sampling

The general goal is to evolve a separate identifier for each of the three minerals being studied. The training scheme requires positive examples (pixels where the target mineral is resident) and negative examples (pixels where the mineral is absent). The solution data is given in a mineral distribution map. This is an RGB bitmap of the Cuprite area whose red, green, and blue channels denote the relative reflectance intensity for AL, KA, and BU respectively. Since these minerals are often mixed throughout the Cuprite site, the RGB channels represent mixed intensities of the minerals.

The majority of the Cuprite area is covered by weak mixtures of the minerals. For example, KA and BU exhibit weak distributions over most of the map. These weak areas will negatively influence evolved results, given both the low intensity of resident spectra and existence of spectra from other incident minerals not being studied. Our hypothesis is that better quality results will be obtained for areas with more intense

Table 1: Training Set Sizes

| | | Threshold | | | | | |
		0.0	0.05	0.15	0.25	0.35	0.50
AL	pos:	40	40	31	26	20	11
	neg:	80	80	89	94	100	109
KA	pos:	75	72	57	46	38	23
	neg:	80	83	98	109	117	132
BU	pos:	77	73	30	19	12	10
	neg:	90	94	137	148	155	157

reflectance values for the minerals of interest. Hence mineral identifiers will be evolved for different *thresholds* of reflectance intensity. Thresholds are used to determine the level of reflectance constituting a positive example. A threshold of 25% means that the reflectance value is considered positive if it has an intensity of at least 25% relative to the maximum reflectance observable (100%). Otherwise, it is treated as a negative instance.

The sizes of the training sets for different thresholds are given in Table 1. Initially, positive and negative example sets were obtained manually, by sampling a diverse selection of pixels throughout the entire map area. The sizes of these sets are listed in the 0% threshold column. The thresholded example sets are refined from these initial sets, by moving positive examples that do not meet the threshold requirements into the negative set. Hence, the positive set sizes decrease as the thresholds are raised.

3.3 GP experiment preparation

Table 2: GP Parameters

Parameter	Value
Population size	1000
Max. generations	100
Max. runs	10
Prob. crossover	0.90
Prob. mutation	0.10
Prob. leaf mutation	0.90
Max. initial depth	2 to 6
Max. depth	17
Tournament size, crossover	4
Tournament size, mutation	7

The GP system used is the typed lilGP 1.1 system (Zongker and Punch 1995). LilGP is a C-based sys-

tem that implements basic tree-oriented GP. Typing is useful since both integer and floating point values are used in evolved programs (Montana 1995). Some self-explanatory GP parameters are given in Table 2. Our use of GP is straight-forward: each program in the population is evaluated on the training examples, and its performance in correctly classifying the examples is measured.

Three target languages are used (Table 3). The languages test spectral properties at single pixel locations of the hyperspectral data. Spatial operators are not used. The spectral operators extract hyperspectral data at a pixel coordinate which is globally set for the current program execution. Language L_1 denotes a boolean decision tree, in which a *true* result means that the target mineral resides at the pixel in question. L_1's relational operators test floating point expressions on hyperspectral image parameters $p[I]$, where I is an index (modulo 168) to the hyperspectral image cube. The floating point function set F is self-explanatory. The *inc* operator increments its integer argument. Ephemeral numbers are randomly generated constants.

L_2 and L_3 are floating point languages, in which evaluated values greater than zero are interpreted as positive identification of the mineral. The L_2 language is the subset of L_1 without boolean expressions. The L_3 language is L_2 supplemented with floating point operators that compute over vectors (contiguous ranges of hyperspectral data). These 2-argument functions compute the minimum, maximum, average, and standard deviation over data vectors. The first argument denotes a starting level in the hyperspectral data. The second argument is evaluated modulo 3, and denotes the depth of the vector: 3, 7, or 11 levels. For example, *vavg(35, 2)* computes the average at the current pixel location for layers 35 through 45 inclusive.

Table 3: Target Languages

Boolean language L_1:
B	::=	(if F<F then B else B)	F<F	*true*	*false*
F	::=	p[I]	F+F	F-F	F*F	F/F	
		min(F,F)	max(F,F)	*ephem flt*
I	::=	inc(I)	*ephem int*

Float language L_2: F, I from L_1

Float language L_3:
$L_2 \cup$ vmin(F,F) | vmax(F,F) | vavg(F,F) | vsdev(F,F)

The fitness value for a program is computed as:

$$Fitness = 1 - \left(\frac{ce}{te} * \frac{cn}{tn} \right)$$

where ce is the number of correctly identified positive examples, te is the total number of positive examples, cn is the number of correctly identified negative examples, and tn is the total number of negative examples. Since the negative training set dwarfs the positive set at higher thresholds, this formula balances positive and negative classification performance with respect to one another.

4 RESULTS

Table 4: Testing and Training Results: % correctly classified pixels

			threshold		
AL	0.05	0.15	0.25	0.35	0.5
testing:					
avg overall	0.825	0.933	0.957	0.966	0.985
best soln	0.875	0.970	0.991	0.995	0.998
TP	0.420	0.587	0.593	0.722	0.644
TN	0.955	0.984	0.997	0.997	0.999
training:					
avg overall	0.87	0.946	0.953	0.961	0.973

			threshold		
KA	0.05	0.15	0.25	0.35	0.5
testing:					
avg overall	0.876	0.952	0.972	0.986	0.987
best soln	0.903	0.964	0.984	0.991	0.994
TP	0.731	0.838	0.869	0.906	0.830
TN	0.963	0.980	0.992	0.997	0.996
training:					
avg overall	0.908	0.963	0.986	0.990	0.966

			threshold		
BU	0.05	0.15	0.25	0.35	0.5
testing:					
avg overall	0.608	0.811	0.972	0.989	0.994
best soln	0.653	0.888	0.993	0.999	0.999
TP	0.797	0.314	0.366	0.592	0.768
TN	0.381	0.945	0.995	1.000	1.000
training:					
avg overall	0.834	0.919	0.986	0.990	0.990

Table 4 shows the training and testing performances for the GP runs. Every mineral and threshold experiment combines the results for 30 runs (3 target languages, 10 runs per language). The testing set is remainder of the input data excluding the training pix-

els. Testing "avg overall" denotes the percentage of correctly classified pixels averaged for all the solutions from the 30 runs. The performance of the single best solution obtained during the 30 runs is given in the "best soln", TP (true positive), and TN (true negative) entries. These values respectively report the percentages of correct pixel classifications for the entire testing image, the positive pixels, and negative pixels. The training "avg overall" is the percentage of positive and negative training examples correctly classified, averaged for all 30 solutions.

The training performance is fairly good amongst all the solutions in the runs, and it improves at higher thresholds. The solution ("best soln") programs obtained higher training performances than the averages reported in the table; in higher threshold cases, the best programs often had 100% training scores.

The testing performance of low-threshold results (especially at 0.05) is marginal. This is due to the noisy reflectance values at that low threshold. The overall testing performance improves at higher thresholds. For the best solutions, TN scores tend to be superior to TP scores, which boosts the overall classification score. The relative abundance of negative training examples compared to positive examples at higher thresholds may explain this.

With AL and KA, the best solutions' TP performance usually improves, while the best TN scores always improve. However, the best TP scores decrease when going to the 0.5 threshold with these minerals, and this was seen with other solutions obtained for these runs. Again, the low number of positive training instances of those minerals at this threshold may explain this (see Figure 1).

For BU, raising the threshold from 0.05 to 0.15, the TP fell from 79.7% to 31.4%. This was seen in most other BU runs as well. The distribution of positive examples of BU decreased dramatically from 73 to 30 examples when moving to the 0.15 threshold, and may not adequately characterize the mineral at this threshold.

Best solutions were distributed fairly evenly amongst the three target languages. L_3 solutions were generally the smallest in terms of tree size, followed by L_1 programs and L_2 programs. L_1 programs were the fastest in wall clock time. L_3 and L_2 solutions were respectively an average of 1.6 and 2.3 times slower than L_1 programs. Overall, runs took between 1 to 20 minutes to complete, with a typical run taking about 6 minutes.

Figure 4 shows classification plots for the best solu-

tions listed in Table 4. In images (a) through (i), grey (TN) and white (TP) are correct classifications, while black denotes erroneous classifications. The classifiers clearly have the most difficulty with the lowest threshold value of 5%. For example, the BU example in (g) only classifies 65.3% of the image correctly. Low-threshold classifiers also varied widely in terms of output characteristics. The classifiers do better at the higher thresholds.

Image (j) deconstructs the classification errors in image (g), by rendering false positives with black, false negatives with white. and the remaining correctly classified pixels as grey. This particular classifier was eager to classify mineral instances, hence its relatively high TP score. Clearly, there is a distribution of BU at 5% and higher throughout a large portion of the map area.

The evolved solution program (in l-expression form) for images (g) and (j) is the following:

```
(- (p (inc 29199))
   (p (inc (inc (inc 23424)))))
```

This simplifies to the expression "p[136] - p[75]". This is using the simple classification rule $p_{2129} > p_{1155}$, where p_{2129} is the pixel reflectance at the $2129nm$ bandwidth. Upon first inspection, this rule does not intuitively correspond to the BU spectra graph in Figure 3, where the BU reflectance at $2129nm$ is lower than at $1155nm$. However, the reflectance chart (k) in Figure 4 shows that this simple relation correctly characterizes BU at this low threshold. The chart was created by finding the average intensity of pixels over the range of spectra used in the testing set, for a constrained area that contained a high density of BU at the 5% threshold. From this graph, it is clear that the relation does in fact accurately classify weak densities of BU. It must be realized, however, that the hyperspectral data at low 5% thresholds are likely poor indicators for any of the minerals studied, given the noise resident at that threshold. In addition, the 1155nm position is near a water vapour absorption feature, and selection of this band by the GP solution may be an artifact of the atmospheric correction procedure. This will be investigated further.

Figure 5 shows the classification expression (simplified from the L_1 source program) for the the best solution for BU at the 50% threshold. The expression uses 12 different frequencies over the entire span of hyperspectral data used.

(a) AL, t = 0.05

(b) AL, t = 0.25

(c) AL, t = 0.50

(d) KA, t = 0.05

(e) KA, t = 0.25

(f) KA, t = 0.50

(g) BU, t = 0.05

(h) BU, t = 0.25

(i) BU, t = 0.50

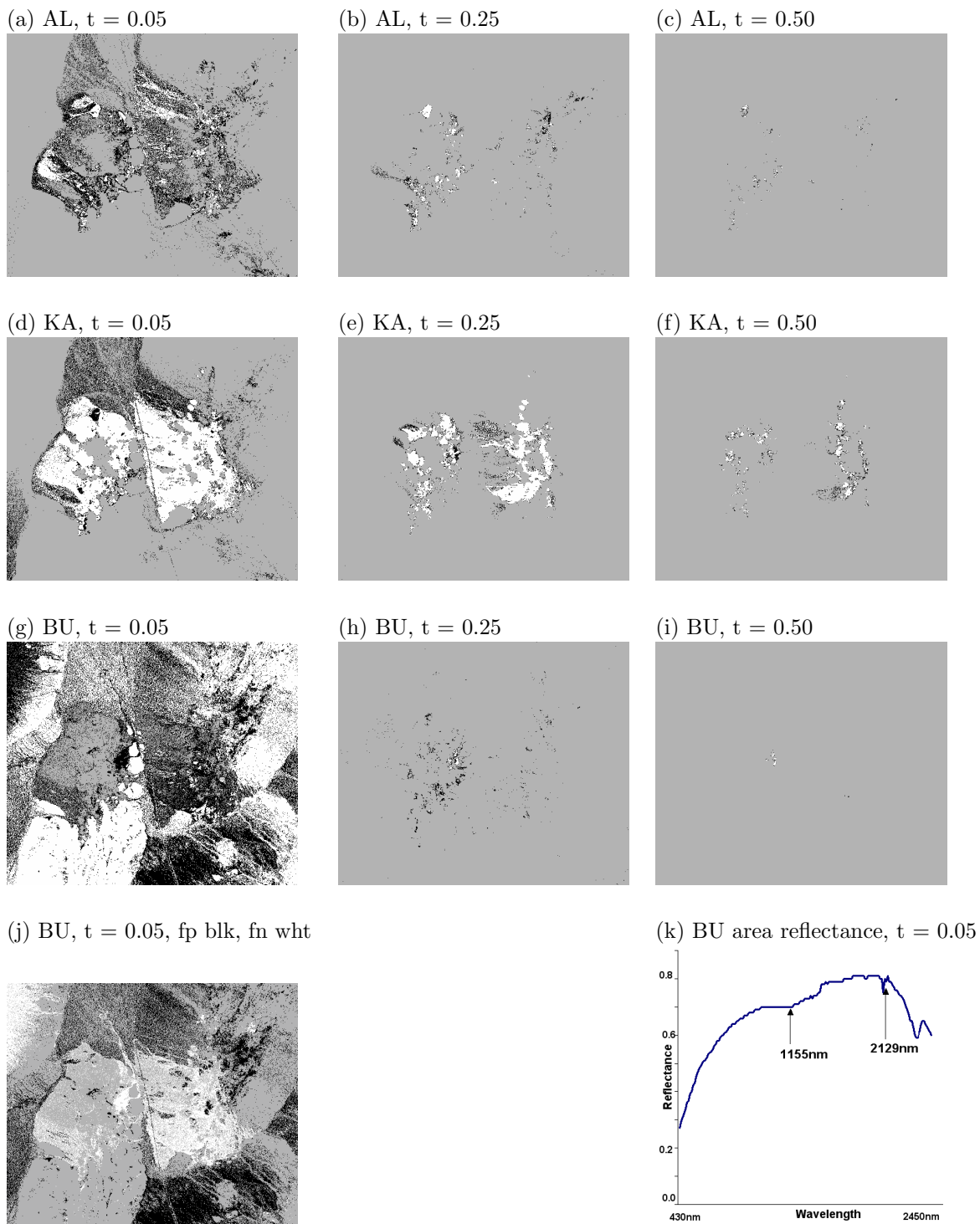(j) BU, t = 0.05, fp blk, fn wht

(k) BU area reflectance, t = 0.05

Figure 4: Classification results for map area: alunite (a-c), kaolinite (d-f), and buddingtonite (g-j). In images (a) through (i), grey is true negative, white is true positive, and black is false negative and false positive. In image (j), grey is both true positive and true negative, black is false positive, and white is false negative.

```
if (max(2.1725 + ((min((p[136] - 1.16407), 0.538215) * 0.19977)), p[88])
              / ((p[30] * 0.19731 / p[163]) + 0.88465))
    < (max(0.76650, p[14]) / min((0.18294 / p[11]), (0.00323 + max(-0.10691, p[74])))))
    then ((-1.1482 / min((p[2] - 0.48504), min(0.70156, p[71]))) < p[31])
    else (p[134] < p[151])
```

Figure 5: Evolved L_1 classifier for BU, 50% threshold

5   CONCLUSION

The hyperspectral mineral identifiers evolved by GP work quite differently from conventional approaches. With least-squares spectra fitting, signature spectra for materials of interest are fitted to the hyperspectral values at each pixel on the map. Identification entails exaggerating the signature differences between materials, and looking for such fluctuations in the hyperspectral data. GP evolves classifiers that find some spectral feature that correctly identifies the existence or absence of a particular mineral. These classifiers do not reference signature spectra, but rather, use the mixed reflectance values as resident in the data set. The success of the classifier depends upon its training performance in differentiating positive and negative examples for the mineral. As a result, material mixtures are automatically accounted for. For example, AL and KA are mixed in a large portion of the Cuprite data set, and the positive training sets for these minerals share many training points.

This implies that the effectiveness of the evolved classifiers implicitly depends upon the context of other materials resident in the geographic area analyzed. The classification logic evolved by GP is best characterized as a function which identifies a mineral in the context of the other minerals resident in the training set. We have not yet tested our mineral identifiers on hyperspectral images from other locations to see how robust these identifiers would be in the presence of materials unseen in the training set. Future work needs to explore the generality of evolved classifiers, in order to see whether a classifier is useful for other geographic locations.

Training set quality is important in our experiments. Our training sets were created by manually selecting positive and negative training points spanning the map area. Although manual sampling is fast and convenient, future work needs to address training sample quality more rigorously. A range of combinations of minerals at various thresholds should be sampled for the positive and negative training sets. This is probably best done via statistical sampling. Such training sets would better represent the varieties of combinations of mineral spectra resident in the images.

Although the fitness formula tries to balance the performance of positive and negative example scoring, many runs produce programs that tend towards being either liberal (eager to identify positive instances) or conservative (eager to report non-instances). Some solutions with very similar fitness scores often have dramatically different classification behaviours, usually falling somewhere on this liberal or conservative dichotomy. These results can mean that the training sets are too small, and evolution is converging prematurely to inadequate solutions.

This work is closest in spirit to that in (Rauss *et al.* 2000). Our L_2 language is similar to theirs, and we also use manually-selected training sets, albeit larger in size than theirs. Their work classified grass from non-grass in hyperspectral images, whereas we classify one of three minerals in each classifier. Our approach can also be compared to the GENIE system (Perkins *et al.* 2000). The GENIE system's application of GP is a bit unusual, as it uses 6 "scratch images", and a fixed-length linear program that may or may not reference these images. Hence the GENIE solution is not as robust a program as a general l-expression program. GENIE also uses conventional classifiers to help analyze and post-process the results from the evolved image analyzer. GENIE uses a large library of spectral and spatial primitive operators, where we use a fairly small set of exclusively spectral operators. When this technology has matured in the future, more careful comparisons between it and other paradigms needs to be undertaken.

References

Aguilar, D.P.K., D.P.M. Cobo, D..R.P. Utrero and D.M.A.H. Nieves (2000). Abundance Extractions from AVIRIS Image Using a Self-Organizing Neu-

ral Network. In: *Proceedings of the Ninth Annual JPL Airborne Earth Science Workshop.*

Brumby, S.P., J. Theiler, S. Perkins, N.R. Harvey and J.J. Szymanski (2001). Genetic programming approach to extracting features from remotely sensed imagery. In: *Proceedings FUSION 2001.*

Civco, D.L. (1993). Artificial neural networks for land-cover classification and mapping. *International Journal of Geographical Information Systems* **7**(2), 173–186.

Clark, R.N. and G.A. Swayze (1995). Mapping Minerals, Amorphous Materials, Environmental Materials, Vegetation, Water, Ice and Snow, and Other Materials: The USGS Tricorder Algorithm. In: *Proceedings of the Fifth Annual JPL Airborne Earth Science Workshop* (R.O. Green, Ed.). pp. 39–40. JPL Publication 95-1.

Daida, J.M., J.D. Hommes, T.F. Bersano-Begey, S.J. Ross and J.F. Vesecky (1996). Algorithm Discovery Using the Genetic Programming Paradigm: Extracting Low-Contrast Curvilinear Features from SAR Images of Arctic Ice. In: *Advances in Genetic Programming II* (P. Angeline and K.E. Kinnear, Eds.). pp. 417–442. MIT Press.

Dreyer, P. (1993). Classification of Land Cover Using Optimized Neural Nets on SPOT Data. *Photogrammetric Engineering and Remote Sensing* **59**(5), 617–621.

Foody, G.M. and M.K. Arora (1997). An evaluation of some factors affecting the accuracy of classification by an artificial neural network. *International Journal of Remote Sensing* **18**(4), 799–810.

Green, R.O., M.L. Eastwood, C.M. Sarture, T.G. Chrien, M. Aronsson, B.J. Chippendale, J.A. Faust, B.E. Pavri, C.J. Chovit, M. Solis, M.R. Olah and O. Williams (1998). Imaging Spectrometry and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS). *Remote Sensing of Environment* **65**, 227–248.

Harvey, N.R., S.P. Brumby, S.J. Perkins, R.B. Porter, J. Theiler, A.C. Young, J.J. Szymanski and J.J. Bloch (2000). Parallel evolution of image processing tools for multispectral imagery. In: *Proceedings Imaging Spectrometry IV: SPIE-4132.* Intl. Soc. for Opt. Eng.. pp. 72–80.

Howard, D. and S.C. Roberts (1999). A Staged Genetic Programming Strategy for Image Analysis. In: *Proc. GECCO-99* (W. Banzhaf *et al*, Ed.). pp. 1047–1052.

Larch, D. (1994). Genetic Algorithms for Terrain Categorization of Landsat Images. In: *Proceedings SPIE-2231: Algorithms for Multispectral and Hyperspectral Imagery.* Intl. Soc. for Opt. Eng.. pp. 2–6.

Merenyi, E., R.B. Singer and W.H. Farrand (1993). Classification of the LCVF AVIRIS Test Site with a Kohonen Artificial Neural Network. In: *Proceedings of the Fourth Annual JPL Airborne Earth Science Workshop.* pp. 117–120.

Montana, D.J. (1995). Strongly Typed Genetic Programming. *Evolutionary Computation* **3**(2), 199–230.

Neville, R.A., C. Nadeau, J. Levesque, T. Szeredi, K. Staenz, P. Hauff and G.A. Borstad (1998). Hyperspectral Imagery for Mineral Exploration: Comparison of Data from Two Airborne Sensors. In: *Proceedings Imaging Spectrometry VI: SPIE-3438.* Intl. Soc. for Opt. Eng.. pp. 74–83.

Perkins, S.J., J. Theiler, S.P. Brumby, N.R. Harvey R.B. Porter, J.J. Szymanski and J.J. Bloch (2000). GENIE: A Hybrid Genetic Algorithm for Feature Classification in Multi-Spectral Images. In: *Applications and Science of Neural Networks, Fuzzy Systems, and Evolutionary Computation III* (B. Bosacchi, D.B. Fogel and J.C. Bezdel, Eds.). pp. 52–62.

Rauss, P.J., J.M. Daida and S. Chaudhary (2000). Classification of Spectral Imagery Using Genetic Programming. In: *GECCO-2000* (D. Whitley *et al.*, Ed.). Morgan Kaufmann. pp. 726–733.

Resmini, R.G., M.E. Jappus, W.S Aldrich, J.C. Harsanyi and M. Anderson (1997). Mineral mapping with Hyperspectral Digital Imagery Collection Experiment (HYDICE) sensor data at Cuprite, Nevada, U.S.A.. *International Journal of Remote Sensing* **18**(7), 1553–1570.

Ridd, M.K., N.D. Ritter, N.A. Bryant and R.O Green (1992). AVIRIS Data and Neural Networks Applied to an Urban Ecosystem. In: *Proceedings of the Second Annual JPL Airborne Earth Science Workshop.* pp. 129–131.

Staenz, K. and D.J. Williams (1997). Retrieval of surface reflectance from hyperspectral data using a look-up table approach. *Canadian Journal Remote Sensing* **23**, 345–368.

Yang, H., F. van der Meer, W. Bakker and Z.J. Tan (1999). A back-propagation neural network for

mineralogical mapping from AVIRIS data. *International Journal of Remote Sensing* **20**(1), 97–110.

Zongker, D. and B. Punch (1995). *lil-gp 1.0 User's Manual*. Dept. of Computer Science, Michigan State University.