

PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Evaluation of image features for discriminating targets from false positives in synthetic aperture sonar imagery

Jeffrey Dale, Aquila Galusha, James Keller, Alina Zare

Jeffrey Dale, Aquila Galusha, James Keller, Alina Zare, "Evaluation of image features for discriminating targets from false positives in synthetic aperture sonar imagery," Proc. SPIE 11012, Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV, 110120A (10 May 2019); doi: 10.1117/12.2519486

SPIE.

Event: SPIE Defense + Commercial Sensing, 2019, Baltimore, Maryland, United States

Distribution Statement A: Approved for public release: distribution is unlimited.

Evaluation of image features for discriminating targets from false positives in synthetic aperture sonar imagery

Jeffrey Dale^a, Aquila Galusha^a, James Keller^a, and Alina Zare^b

^aUniversity of Missouri, Columbia, MO

^bUniversity of Florida, Gainesville, FL

ABSTRACT

With the increasing popularity of using autonomous underwater vehicles (AUVs) to gather large quantities of Synthetic Aperture Sonar (SAS) seafloor imagery, the burden on human operators to identify targets in these seafloor images has increased significantly. Existing methods of automated target detection can have perfect or near-perfect accuracy, but often produce a high ratio of false positives. Thus, it is desired to find features that discriminate between targets and high-confidence false alarms. In this paper, we examine the potential of several classical methods of feature extraction in how well their generated features can separate the two classes of image tiles: those containing targets from those containing no targets. To quantify the ability of a set of features to separate these classes, we measure the region-based cross validation accuracy of a linear SVM trained on the features in question, extracted from SAS imagery provided to us by the U.S. Navy. We show that these general feature extraction methods show potential in the ATR problem, suggesting further research is warranted.

Keywords: Feature Extraction, Synthetic Aperture Sonar, Supervised Learning, Automated Target Recognition

1. INTRODUCTION

The inherent high dimensionality of images has always posed an issue for classification, leading to extensive research in dimensionality reduction. Classical techniques in this area, which preceded the push for image-based classification, include principal component analysis, linear discriminant analysis, and non-negative matrix factorization.¹ The 1990s and early 2000s gave rise to a number of approaches to dimensionality reduction tailored specifically to the image classification problem, such as Histogram of Oriented Gradients (HOG)² and Local Binary Patterns (LBP).³

As computers became more powerful, neural network-based approaches to image classification have saturated the literature, especially convolutional neural networks (CNNs).⁴ We have seen applications of CNNs for explosive hazard detection in synthetic aperture sonar (SAS) imagery^{5,6} as well as in synthetic aperture radar (SAR) imagery.⁷ At the intersection of feature-based classification and neural network-based classification is the autoencoder,⁸ a neural network designed for unsupervised dimensionality reduction.

The goal of this survey is to gauge the effectiveness of classical feature-based approaches in classifying regions of SAS images as containing a target or benign. A parallel study by Galusha et al.⁵ examines the use of deep CNNs for classification on the same SAS dataset.

Further author information: (Send correspondence to J.D.)

J.D.: E-mail: jjdale@mail.missouri.edu

A.G.: E-mail: apgqpc@mail.missouri.edu

J.K.: E-mail: kellerj@missouri.edu

A.Z.: E-mail: azare@ece.ufl.edu

2. BACKGROUND

2.1 Features

The focus of this work is in which image features provide the best discriminative information between targets and false alarms, and as such, we provide a description of each feature used in the experiments to follow. The features which are easy to visualize, namely HOG, LBP, and Lacunarity are shown in Figure 1. The other features do not naturally lend themselves to image-like representations. Table 1 compares each feature on its invariance to common transformations. Descriptions of the features that follow are ordered chronologically by initial publication date.

2.1.1 Hu Moments

Just as probability distributions can be uniquely characterized by their moments, Hu asserts that the same idea can be applied to two dimensional textures.⁹ Since proving equivalence of two distributions by moments requires either knowledge of the moment generating function or an infinite number of moments, Hu proposes a finite set of seven moments, called the Hu moment invariants, that are capable of accurately characterizing many textures. These moments, defined in Equations 61-62 of Hu's paper,⁹ are invariant to translation, scale, and rotation. We compute these moments across blocks of each image and concatenate the result to form our feature vector.

2.1.2 Haralick Features

The features proposed by Haralick are based on the gray-level co-occurrence matrix¹⁰ (GLCM, also called gray-tone spatial dependence matrix) of an image, whose entries (i, j) represent the number of times a pixel of intensity i appears distance d away from a pixel of intensity j . This distance is two-dimensional relative to the currently examined pixel, and is often represented as either an (x, y) offset in Cartesian coordinates or an angle and distance (ρ, θ) in polar coordinates. Before computing the GLCM, the image must be quantized into a given number of levels q . The resulting GLCM is of size $q \times q$. For each angle and distance specified, 14 measures are computed on the GLCM,¹¹ specified in Appendix I of Haralick's paper.¹⁰ The angles used are 0° , 45° , 90° , and 135° ; distances are a user-specified parameter. Taking the mean and range of each of the 14 measures across all angles gives 28 features per distance.

2.1.3 Zernike Features

In 1934, Zernike developed a set of complex polynomials for use in optics.¹² These polynomials are orthogonal over the unit disk, and thus contain are efficient representations of underlying features. It wasn't until the 1990s that these polynomials were applied image feature extraction. By treating an image as a two-dimensional function and projecting onto the orthogonal Zernike polynomial basis, a set of rotationally invariant features can be extracted.¹³ Another advantage of using an orthogonal polynomial basis is ease of reconstruction, accomplished by a simple product. With proper image normalization, these features can also achieve scale and translation invariance.

2.1.4 Local Binary Patterns

Local Binary Patterns (LBP) are a powerful feature descriptor achieving invariance to change in both luminance and rotation.³ P uniformly spaced points g_p ($p = 0, \dots, P - 1$) at radius R from a center point g_c are examined, with intensity values interpolated as necessary. In order to be invariant to luminance changes, each of these points is represented as $\mathbb{I}(g_p \geq g_c)$, where \mathbb{I} is the indicator function. Concatenating these indicator values yields a binary number. For rotation invariance, the minimum result of repeated circular bitwise right shifts on the binary number yields the LBP value. Finally, after partitioning the image into cells, a histogram is computed for each cell consisting of LBP values residing in that cell. The concatenation of these histograms forms the LBP features for the image.

Feature	Transformation			
	Translation	Rotation	Scale	Luminance
HOG	\times^1	\times^1	\times	✓
LBP	\times^1	✓	\times^2	✓
Hu Moments	✓	✓	✓	✓
Haralick	✓	\times^2	\times^2	✓
Lacunarity	+	\times	\times^2	✓ ³

¹Change in feature vector small if magnitude of transformation is small.

²Scale parameter exists, making feature equivariant in scale-space.

³Invariant with minor adjustment, see Appendix.

Table 1: The invariance of each tested feature to translation, rotation, scale, and luminance. Check marks (✓), crosses (+), and x marks (\times) represent invariance, equivariance, and neither invariance nor equivariance, respectively.

2.1.5 Histogram of Oriented Gradients

Originally motivated by the problem of detecting pedestrians in images,² histogram of oriented gradients (HOG) features have also proven quite effective in face recognition,¹⁴ hand gesture recognition,¹⁵ and in landmine detection.¹⁶ To compute HOG features, the image in question is partitioned into *cells* of a given size with each cell belonging to potentially overlapping *blocks* of nearby cells. For each cell, gradients are computed at each pixel using the centered derivative mask $[-1 \ 0 \ 1]$ and normalized based on the gradients of other cells in the same block. The resulting values are then binned into a histogram. The concatenation of all such histograms form the HOG feature. For images with multiple color channels, such as RGB images, HOG features are computed independently for each channel and concatenated.

2.1.6 Lacunarity

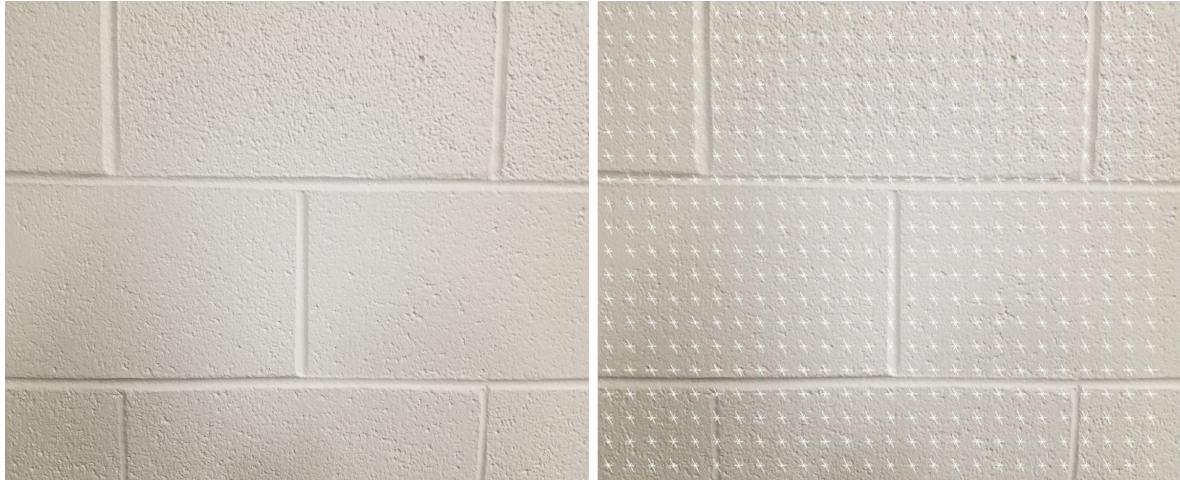
Lacunarity (Latin: “lacuna” meaning “gap”) is a measure originating in fractal geometry that describes the size and frequency of gaps in a texture.¹⁷ A texture that is highly lacunar will have many and/or large gaps where the variance of the texture is small. Lacunarity analysis has shown to be useful not only in describing fractal features, but nonfractal features in general spatial patterns as well.¹⁸ Methods of computing lacunarity often involve gliding box algorithms or box counting algorithms,¹⁹ however these approaches tend to be computationally expensive when image size is large, as in SAS imagery. Inspired by lacunarity, a simplified sliding window definition was proposed using integral images to quickly characterize textures,²⁰ defined by

$$L(i, j) = \frac{\sigma_R^2}{\mu_R^2} \quad (1)$$

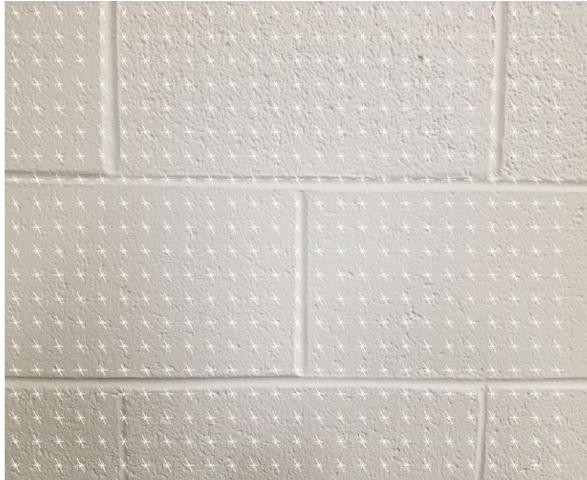
The subscript R denotes the set of pixels in a box of a given size centered at pixel (i, j) , with σ_R^2 representing the variance of these pixels in this box and μ_R the mean. The associated feature for this measure is a vector of the mean lacunarity value across the image for varying window sizes.

2.2 Support Vector Machines (SVMs)

Dating as far back as the 1960s,²¹ SVMs began as linear classifiers designed to separate data using a maximum-margin hyperplane. Limitations of early SVMs included their inability to successfully learn decision boundaries when classes are not linearly separable. Rather than extending SVMs to separate non-linearly separable data, a so-called “kernel trick” was proposed to transform the data into a space where it is linearly separable.²² Common kernels for use in modern SVMs include linear kernels, Gaussian kernels, and polynomial kernels. Based on ease of implementation and on the time required to train, we chose to use linear kernel SVMs as the classifier in our experiments.



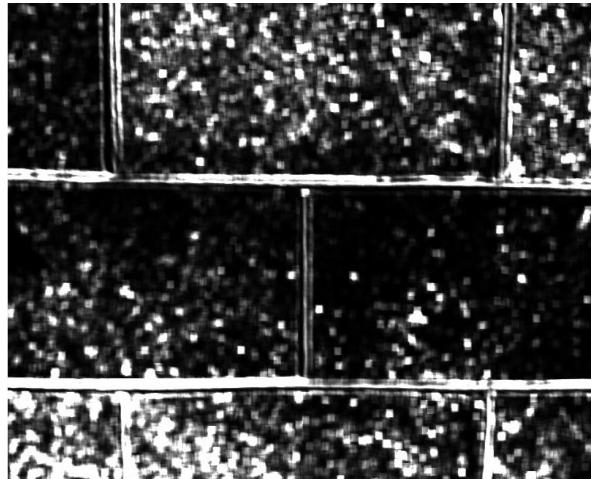
(a) Original image of a brick wall.



(b) Gradients from HOG.



(c) LBP response at each pixel.



(d) Lacunarity at each pixel.

Figure 1: Visualization of HOG, LBP, and Lacunarity features applied to a picture of a brick wall. HOG was run using a cell size of 30×30 , block size of 4×4 , and 3 histogram bins. LBP used 8 neighbors at radius 1 with nearest neighbor interpolation. Lacunarity used a window size of 10×10 . The other features considered are not easily visualized.

3. DATASET

Our experiments in feature-based image classification were all performed on a set of SAS image tiles centered around hits from the combined RX detector²³ after suppression of low confidence values. Each of these tiles are of size 201×301 , encompassing 100 pixels to the left of the hit, 200 pixels to the right, and 100 pixels above and below the hit. This configuration, designed to capture the target and its shadow, is illustrated in Figure 2. For hits that are near the border of the image, the image is padded with zeros so that the tile remains 201×301 in size.

It should also be noted that targets are very sparse in this dataset; more than 99% of tiles are benign. This class imbalance should be taken into consideration when viewing accuracy measurements, as even the trivial classifier that always reports “not a target” will attain upward of 99.9% accuracy. For this reason, we focus on the false positive and false negative rates (FPR and FNR).

4. PARAMETER TUNING

The choice of parameters for a feature extraction algorithm is crucial in how well that feature describes the data. With the goal of designing a completely fair experiment, the parameters for each feature were tuned using a global optimization algorithm. However, seeing as we can often find several parameter sets all achieving 100% accuracy on our training set, we devise a rudimentary separability measure to quantify how well a particular set of parameter separates feature space. We use this measure in the global search heuristics applied for parameter selection.

4.1 Separability Score

The goal of an SVM is to find a maximum-margin hyperplane that separates two classes. By simply examining the width of the margin, we can quantify the separability of a feature space produced by a set of feature parameters. The ideal feature would create a feature space with a very large margin, which would, in theory, be more robust to noise in new samples for classification. Figure 3 illustrates our reasoning.

We must also consider the case in which the classes are not separable, which makes the width of the margin uninformative. If this is the case, our score should be based on the classification accuracy. Since soft-margin SVMs are more suited to inseparable data, we use accuracy of a soft-margin SVM. Combining the ideas of margin-based scoring and accuracy-based scoring, we arrive at the proposed *separability score*, given by

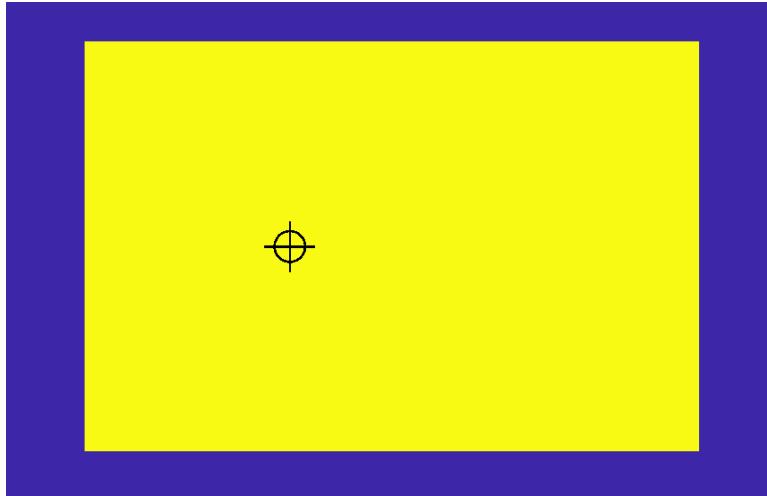


Figure 2: Configuration of an image tile (yellow) around prescreener hit (black marker). Dark blue indicates background image that is not considered. The yellow rectangular area is defined by 100 pixels above, below, and left of the hit and 200 pixels right of the hit.

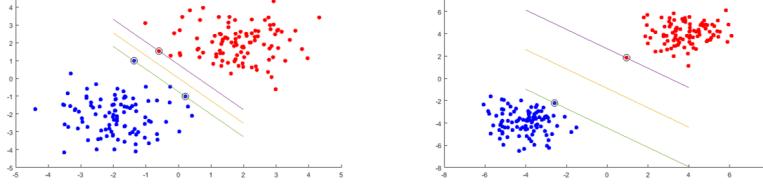


Figure 3: Each image shows two perfectly separable two-dimensional Gaussian random samples and the margin found by a hard-margin SVM. The width of the margin is larger on the second, more separable image.

$$Sep(X, \vec{d}) = \begin{cases} \log A_s(X, \vec{y}) & 0 \leq A_h(X, \vec{y}) < 1 \\ \frac{2}{\|w_h\|} & A_h(X, \vec{y}) = 1 \end{cases} \quad (2)$$

where

- X is an $n \times d$ matrix of n feature points in d -space,
- $y_i \in \{-1, 1\}$ is the label for the i th sample,
- $A_h(X, \vec{y})$ is the accuracy of a hard-margin SVM classifier on the data,
- $A_s(X, \vec{y})$ is the accuracy of a soft-margin SVM classifier on the data, and
- $2/\|w_h\|$ is the width of the margin in a hard-margin SVM with weights w_h .

This formulation of a separability score has two desirable properties. First, the range is \mathbb{R} , making it convenient for use as an objective function in optimization. Maximizing $Sep(X, \vec{d})$ with respect to X corresponds to finding a maximally separable feature space. Second, as a convenience factor, the sign of Sep indicates whether the classes are separable. In our parameter tuning experiments, we minimize $-Sep(X, \vec{d})$ with respect to the feature parameters that generated X .

Unfortunately, evaluating this function is computationally expensive, given that we potentially need to train two SVMs. Another drawback the separability score's lack of translation between different problems. Since it is based on the performance of the SVM in separating the known classes, separability score can vary wildly with dimensionality of feature space and number of training samples. While it should not be compared across different situations, we believe that it works well for our case of finding Pareto-optimal parameter sets on a single training set at a time.

4.2 Optimization Methods

Motivated by the desire for both high separability and low dimensionality in feature space, our optimization methods incorporate techniques from the domain of multi-objective optimization. We search for Pareto-optimal solutions in terms of dimensionality and separability score; however, the search is guided only by separability score. The global search algorithms we used are variable neighborhood search (VNS)²⁴ for discrete/integer optimization problems and a standard genetic algorithm (GA)²⁵ for combinatorial optimization. Implementation-specific information about how the parameters for each feature were optimized is shown in Table 2, and the Pareto-optimal parameter sets for each feature are listed in the Appendix.

5. EXPERIMENTS

To ease the severe class imbalance of this SAS dataset, all training and parameter tuning takes place on a subset of the dataset consisting of all targets and three false alarms per target. We refer to this subset as the library tiles. As a pleasant consequence, training is far quicker on the library tiles than it would be on the whole dataset.

Five-fold cross validation was used for testing, where each fold corresponds to the SAS images collected from one geographic location. Table 3 shows the distribution of images and tiles across locations and Figure 4 shows the breakdown from all SAS images to the library tiles. Fold n in cross validation refers to testing on location n and training on all locations except n . Parameter tuning was performed in the same cross validation style, where parameters are tuned on the labeled samples in all locations except n . Location n is not seen in parameter

Feature	Optimization Type		Parameters	# Variables	Method
	Discrete	Combinatorial			
HOG	✓	✗	CS, BS, BO, # Bins	7 integers	VNS
LBP	✓	✗	Rad., CS, # Neighbors	4 integers	VNS
Hu Moments	✓	✗	BS	2 integers	VNS
Haralick	✓	✗	# Levels, Dist, BS	4 integers	VNS
Lacunarity	✗	✓	Window Sizes	200 bits	GA
Zernike	✓	✗	BS, Order	3 integers	VNS

Acronyms: CS - Cell Size, BS - Block Size, BO - Block Overlap.

Table 2: Details about how feature parameters were optimized for each feature tested.

	Location				
	Loc. 1	Loc. 2	Loc. 3	Loc. 4	Loc. 5
Total Images	620	2032	256	298	144
Total Tiles	244756	802088	100377	117553	56770
Viable Tiles	27105	103625	27604	26194	14379
Library Tiles	196	596	132	104	100

Table 3: Information about the five location folds used in cross validation experiments. Library tiles are used for training/tuning and viable tiles are used for testing.

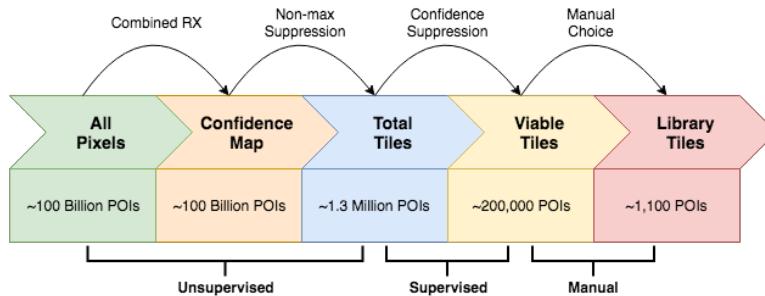


Figure 4: Visual description of tiers of the SAS dataset. Each component is a subset of the tier to its left and a superset of the tier to its right. The confidence suppression method is an adaptive filter that operates on combined RX confidences, and is a work in progress. Note: POI = point of interest, corresponding to one tile to examine.

tuning on fold n . This process emulates a series of blind tests as would be the case in real world missions, and explains why we have five parameter sets for each feature.

To obtain what we believe to be an optimal set of features for each fold, we performed the following.

1. **Individual Tuning.** For each feature, find Pareto optimal parameters with respect to separability score and feature size using labeled tiles from current fold of library set. Manually select (at most) three desirable parameter sets that provide the best balance between feature size and separability score.
2. **Combination Tuning.** Perform exhaustive search of all unique combinations of individually selected features (concatenated to form a single feature) to obtain Pareto optimal combinations. If exhaustive search is not tractable, sequential feature selection can be used in place of exhaustive search.
3. **Final Feature Set Selection.** Choose the combination of features at the knee of the Pareto plot, yielding one set of features per fold.

After obtaining one set of features per fold, we evaluate the performance of each fold's feature set on the testing set for that fold. Measures of interest primarily include false positive rate (FPR) and false negative rate (FNR). A ROC curve based on SVM prediction confidence is also shown for each fold.

6. RESULTS

As results for individual parameter tuning are quite lengthy, and likely not of interest to the average reader, they are shown in the appendix.

The set of Pareto optimal feature combinations are plotted in Figure 5. Each point in a scatter plot represents one combination of features. It is immediately apparent that the Pareto fronts shown in Figure 5 are very steep, nearly L-shaped in most cases. This indicates that there is one point (combination of features) that nearly dominates the rest, residing at the “knee” of the Pareto front. Using this process, we chose one combination of features for each fold, indicated by the red star shape in each plot.

Figure 6 shows the composition of the chosen feature combinations for each fold. We immediately notice that Zernike moments severely outperform the other five features, making up the entire feature for all folds except Fold 4, which also includes Haralick features. Examining the individual tuning results in the Appendix, Tables 15 and 13 affirm that Haralick features and Zernike moments have significantly higher separability scores than the other tested features. In addition, the relatively small feature size of Zernike moments makes it easier for the SVM to find a good decision boundary, yielding higher separability score.

The consistent ranking of Zernike moments by itself over any combination of tested features indicates that the extra information provided by different features serves mostly as noise to the SVM, yielding little to no improvement in separability score. In many cases, the separability score was severely diminished by combining Zernike moments with other features, especially those with a large feature size.

The true test of how well these features perform is not in how well it satisfies a fitness function, but in how well the features perform on our blind test for each fold. Figure 7 details the results of these blind tests. At 5% false positive ratio, all locations except Location 2 achieve above 85% detection. The performance on Location 2 is clearly attributed to the size of its training set; less than half the size of the other locations. The “average” curve Figure 7a shows all confidences from all locations put together, and gives us insight into the overall performance of these feature sets. Figure 7b omits Location 2 for lack of sufficient training data, and adjusts the average curve accordingly. When considering all locations, the average curve attains 100% detection at around 16% FPR, and at 5% FPR, achieves around 80% detection. Upon removing Location 2, over 90% of targets have an SVM confidence of one. We attain 100% detection at around 22% FPR on average when considering Locations 1 and 3-5, and 5% FPR corresponds to just over 90% detection.

The confusion matrices in Figure 7c further break down the performance of the classifier trained on features resulting from parameter tuning. We are primarily concerned with false negative rate (FNR), that is, cases where the classifier called a region benign that contained a target. Achieving FNR of 0%, or very near 0%, is crucial

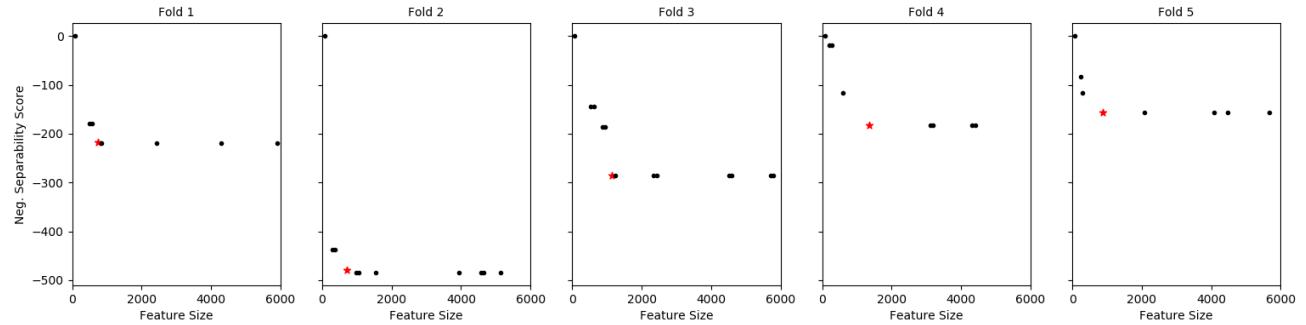


Figure 5: Pareto optimal combinations of individually tuned features for each fold. The red star indicates the fitness values of our final chosen set of features. The composition of these features are shown in Figure 6

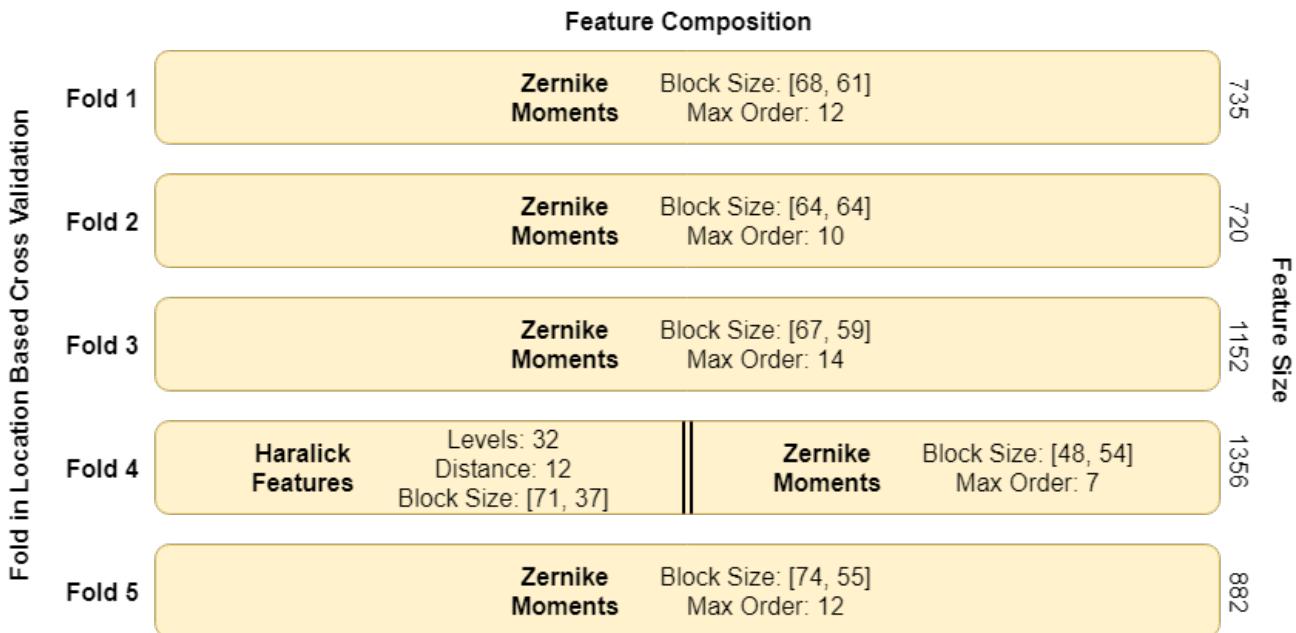
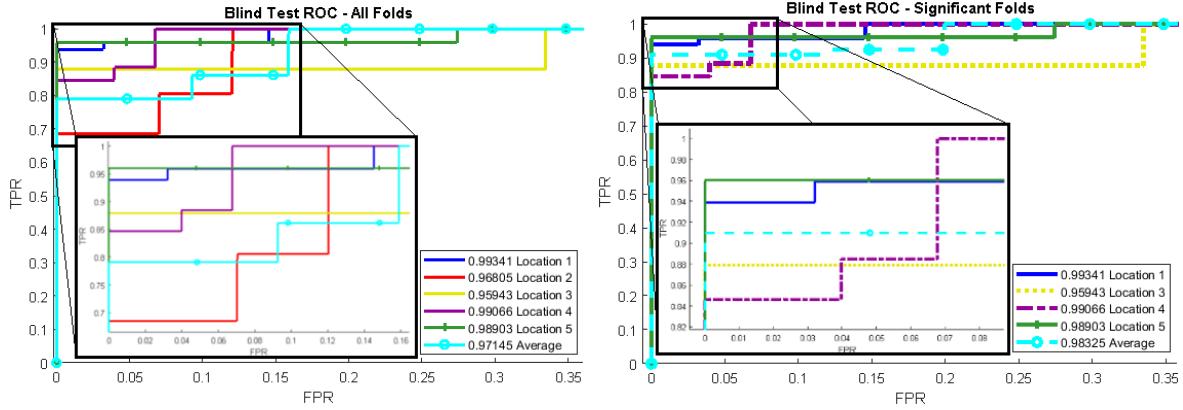
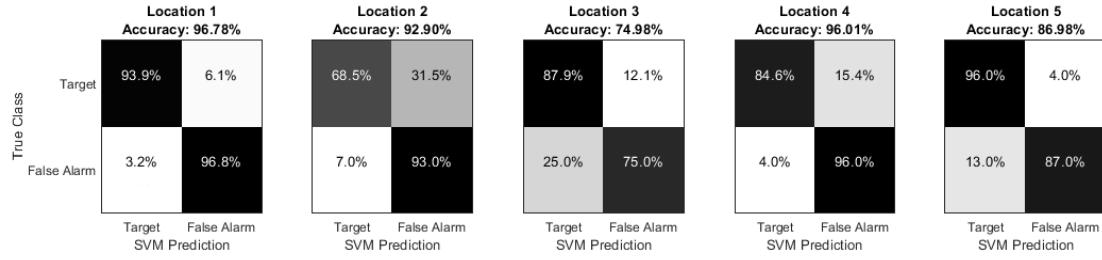


Figure 6: The specific composition of the final feature set for each fold after parameter tuning. The fitness values of these feature sets are shown in the context of other Pareto optimal feature sets in Figure 5.



(a) ROC showing false positive ratio at each location, (b) Same ROC curve, omitting Location 2 for lack of along with concatenation of all confidences.



(c) Confusion matrix of each fold.

Figure 7: Results from our simulated blind tests using final chosen image features.

in explosive hazard detection. We may sometimes wish to trade a small increase in FNR for a large decrease in FPR, but in general, flagging a target as benign should happen very rarely if at all. For locations 1, 3, 4, and 5, FNR was relatively low. Location 2 demonstrated a failure of the tuned features to effectively discriminate targets from false alarms. In fairness, the training set for location 2 was significantly smaller than for the other locations, however such poor performance raises concern about whether these features can generalize to new and vastly different terrain. A general framework to adapt to changing environments, dubbed environmentally aware automated target recognition (EAATR), is an area of active research.

7. CONCLUSION

We have presented a thorough survey of feature-based approaches to image classification. Since the parameters of image features can significantly affect the ability of the features to separate classes, we put forth substantial effort into finding the best parameters for each feature, tailored to SAS imagery.

As different features can capture different characteristics of the images, we exhaustively searched every combination of individually optimal features (with respect to feature size and separability score). All features were classified as target or benign across a 5-location cross validation approach to simulate a blind test as would occur in the field.

We found that Zernike moments best optimized our fitness function (separability score) in parameter searching, so much so that Zernike moments by itself almost always yielded a higher separability score than in combination with any other feature. Haralick features also performed well according to separability score.

While the results presented here are not competitive with the current state of the art in ATR, performance so far indicates that this approach has potential. We believe that by tailoring the features to the SAS data, we could see significant improvement in FNR and FPR. Some future work includes taking fine features near the center of the image and sparse features around the edges, including both high frequency and low frequency

images in our feature sets, and incorporating classifiers that support non-linearly separable data. By taking more than one point on the Pareto front of feature combinations, we allow the potential for classifier fusion.

We stress that, unlike deep convolutional neural network (DCNN) approaches, feature-based approaches to classification allow us to insert human intuition into characteristics of targets and false alarms. In this way, a human is likely to understand why a classifier made the decision it did, one step closer to explainable AI.

ACKNOWLEDGMENTS

This work was funded by the Office of Naval Research grant number N00014-161-2323 to support the U.S. Naval Surface Warfare Center.

REFERENCES

- [1] Nielsen, F. and Bhatia, R., [*Matrix information geometry*], Springer (2013).
- [2] Dalal, N. and Triggs, B., “Histograms of oriented gradients for human detection,” in [*Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*], 1, 886–893, IEEE (2005).
- [3] Ojala, T., Pietikainen, M., and Maenpaa, T., “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on pattern analysis and machine intelligence* **24**(7), 971–987 (2002).
- [4] Krizhevsky, A., Sutskever, I., and Hinton, G. E., “Imagenet classification with deep convolutional neural networks,” in [*Advances in neural information processing systems*], 1097–1105 (2012).
- [5] Galusha, A., Keller, J., and Zare, A., “Deep convolutional neural network target classification for underwater synthetic aperture sonar imagery,” in [*Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIV*], (2019).
- [6] Williams, D. P., “Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks,” in [*Pattern Recognition (ICPR), 2016 23rd International Conference on*], 2497–2502, IEEE (2016).
- [7] Ding, J., Chen, B., Liu, H., and Huang, M., “Convolutional neural network with data augmentation for sar target recognition,” *IEEE Geoscience and remote sensing letters* **13**(3), 364–368 (2016).
- [8] Hinton, G. E. and Salakhutdinov, R. R., “Reducing the dimensionality of data with neural networks,” *science* **313**(5786), 504–507 (2006).
- [9] Hu, M.-K., “Visual pattern recognition by moment invariants,” *IRE transactions on information theory* **8**(2), 179–187 (1962).
- [10] Haralick, R. M., Shanmugam, K., et al., “Textural features for image classification,” *IEEE Transactions on systems, man, and cybernetics* (6), 610–621 (1973).
- [11] Monzel, R., “haralicktexturefeatures (<https://www.mathworks.com/matlabcentral/fileexchange/58769-haralicktexturefeatures>),” in [*MATLAB Central File Exchange*], (2019).
- [12] Zernike, F., “Beugungstheorie des schneidenverfahrens und seiner verbesserten form, der phasenkontrast-methode,” *Physica* **1**(7-12), 689–704 (1934).
- [13] Khotanzad, A. and Hong, Y. H., “Invariant image recognition by zernike moments,” *IEEE Transactions on pattern analysis and machine intelligence* **12**(5), 489–497 (1990).
- [14] Déniz, O., Bueno, G., Salido, J., and De la Torre, F., “Face recognition using histograms of oriented gradients,” *Pattern Recognition Letters* **32**(12), 1598–1603 (2011).
- [15] Misra, A., Abe, T., and Deguchi, K., “Hand gesture recognition using histogram of oriented gradients and partial least squares regression.,” in [*MVA*], 479–482 (2011).
- [16] Torrione, P. A., Morton, K. D., Sakaguchi, R., and Collins, L. M., “Histograms of oriented gradients for landmine detection in ground-penetrating radar data,” *IEEE Transactions on Geoscience and Remote Sensing* **52**(3), 1539–1550 (2014).
- [17] Mandelbrot, B. B., [*The fractal geometry of nature*], vol. 1, WH freeman New York (1982).
- [18] Plotnick, R. E., Gardner, R. H., Hargrove, W. W., Prestegard, K., and Perlmutter, M., “Lacunarity analysis: a general technique for the analysis of spatial patterns,” *Physical review E* **53**(5), 5461 (1996).

- [19] Keller, J. M., Chen, S., and Crownover, R. M., “Texture description and segmentation through fractal geometry,” *Computer Vision, Graphics, and image processing* **45**(2), 150–166 (1989).
- [20] Williams, D. P., “Fast unsupervised seafloor characterization in sonar imagery using lacunarity,” *IEEE transactions on Geoscience and Remote Sensing* **53**(11), 6022–6034 (2015).
- [21] Vapnik, V., “Pattern recognition using generalized portrait method,” *Automation and remote control* **24**, 774–780 (1963).
- [22] Boser, B. E., Guyon, I. M., and Vapnik, V. N., “A training algorithm for optimal margin classifiers,” in [*Proceedings of the fifth annual workshop on Computational learning theory*], 144–152, ACM (1992).
- [23] Galusha, A., Galusha, G., Keller, J., and Zare, A., “A fast target detection algorithm for underwater synthetic aperture sonar imagery,” in [*Detection and Sensing of Mines, Explosive Objects, and Obscured Targets XXIII*], **10628**, 106280Z, International Society for Optics and Photonics (2018).
- [24] Mladenović, N. and Hansen, P., “Variable neighborhood search,” *Computers & operations research* **24**(11), 1097–1100 (1997).
- [25] Brownlee, J., [Clever algorithms: nature-inspired programming recipes], Jason Brownlee (2011).

APPENDIX

For the sake of transparency and reproducible research, this section contains specific settings used in the experiments presented, and may not be of interest to most readers.

Optimization Hyperparameters

For all parameter tuning experiments, feature size was limited to 6000, roughly 10% of the number of pixels in each (201×301) image.

HOG	Parameter			
	Cell Size	Block Size	Block Overlap	Num. Bins
Bounds	[2, 64]	[1, 16]	[0, 16]	[3, 9]
Neighborhoods	{4, 8, 32}	{1, 2, 4}	{1, 1, 2}	{1, 2, 3}
Domain	\mathbb{Z}^2	\mathbb{Z}^2	\mathbb{Z}^2	\mathbb{Z}
Initial Guess	(32, 16)	(2, 2)	(1, 1)	3
Search Space	3969	256	289	7

Constraints: $BO < BS$ $BS \times CS \leq IS$
Cardinality of Search Space: $\approx 2 \cdot 10^9$

Table 4: Settings for tuning HOG parameters using VNS. (BS - Block Size, BO - Block Overlap, IS - Image Size)

LBP	Parameter		
	Neighbors	Radius	Cell Size
Bounds	[4, 24]	[1, 5]	[2, 64]
Neighborhoods	{4, 8, 16}	{1, 2, 3}	{4, 8, 32}
Domain	\mathbb{Z}	\mathbb{Z}	\mathbb{Z}^2
Initial Guess	16	5	[64, 24]
Search Space	21	5	3969

Constraints: None
Cardinality of Search Space: $\approx 4 \cdot 10^5$

Table 5: Settings for tuning LBP parameters using VNS.

Haralick	Parameter		
	Num. Levels	Distance	Block Size
Bounds	[2, 32]	[1, 12]	[2, IS]
Neighborhoods	{4, 8, 16}	{1, 2, 3}	{4, 8, 32}
Domain	\mathbb{Z}	\mathbb{Z}	\mathbb{Z}^2
Initial Guess	12	8	[64, 32]
Search Space	31	12	60000

Constraints: None

Cardinality of Search Space: $\approx 4 \cdot 10^5$

Acronyms: IS - image size (201×301)

Table 6: Settings for tuning Haralick parameters using VNS.

Hu Moments	Parameter	
	Block Size	
Bounds	[8, IS]	
Neighborhoods	{4, 8, 16}	
Domain	\mathbb{Z}^2	
Initial Guess	(16, 16)	
Search Space	60000	

Constraints: None

Cardinality of Search Space: $6 \cdot 10^4$

Acronyms: IS - image size (201×301)

Table 7: Settings for tuning Hu Moments parameters using VNS.

Zernike Moments	Parameter	
	Block Size	Order
Bounds	[32, IS]	[0, 20]
Neighborhoods	{4, 8, 16}	{2, 4, 6}
Domain	\mathbb{Z}^2	\mathbb{Z}
Initial Guess	(64, 64)	10
Search Space	45461	21

Constraints: None

Cardinality of Search Space: $\approx 10^6$

Acronyms: IS - image size (201×301)

Table 8: Settings for tuning Zernike Moments parameters using VNS. The “order” parameter indicates that the feature is composed of all Zernike moments of equal or smaller order than the parameter value.

Lacunarity	Variable	Hyperparameters			
		Window Sizes	Pop. Size	Cross. Rate	Mut. Rate
	$\subseteq \{2, 3, \dots, 201\}$	50	0.9	0.1	
Stopping Criteria: 1000 Function Evaluations					
Cardinality of Search Space: $2^{200} \approx 1.6 \cdot 10^{60}$					

Table 9: Settings for tuning LBP parameters using a genetic algorithm. Windows are constrained to be square, as the search space was too large when trying to tune two window size parameters.

Chosen Parameter Sets

When possible, we tried to choose three Pareto-optimal parameter sets for each feature, near the minimum, mean, and maximum feature size in the Pareto plot. In cases where the feature size was very small or all solutions had similar feature sizes, we chose the single solution that attained the best separability score.

HOG	Parameter Set				Score	
	Cell Size	Block Size	Block Overlap	Num. Bins	Sep. Score	Feat. Size
Fold 1	(35, 15)	(3, 2)	(2, 1)	9	0.0698	1140
	(37, 9)	(3, 2)	(2, 1)	9	0.1642	3456
	(33, 4)	(3, 3)	(0, 1)	9	0.2365	5994
Fold 2	(33, 14)	(3, 2)	(2, 1)	5	0.1504	1200
	(33, 13)	(3, 2)	(2, 1)	6	0.2653	3168
	(33, 11)	(3, 2)	(2, 1)	9	0.3447	5616
Fold 3	(33, 14)	(3, 3)	(2, 2)	4	0.0879	1440
	(33, 12)	(3, 3)	(2, 2)	7	0.1364	3024
	(29, 12)	(3, 3)	(2, 2)	7	0.1966	5796
Fold 4	(32, 13)	(2, 2)	(1, 1)	4	0.0569	1056
	(30, 11)	(2, 3)	(1, 2)	6	0.1337	2880
	(29, 6)	(1, 5)	(0, 3)	8	0.2388	5520
Fold 5	(26, 19)	(2, 3)	(1, 1)	8	0.0609	1008
	(25, 15)	(2, 2)	(1, 1)	7	0.1540	3724
	(17, 17)	(2, 2)	(1, 1)	9	0.2118	5760

Table 10: Chosen Pareto-optimal parameters for HOG on each of five cross validation folds.

LBP	Parameter Set				Score	
	Num.	Neighbors	Radius	Cell Size	Sep. Score	Feat. Size
Fold 1	6		5	(25, 12)	0.2603	1400
	6		5	(12, 11)	0.5014	3024
	6		3	(8, 10)	0.7826	6000
Fold 2	6		5	(40, 25)	0.1944	480
	6		5	(11, 12)	0.8265	3150
	9		4	(12, 9)	0.9343	5808
Fold 3	6		3	(32, 12)	0.2140	1050
	6		3	(10, 14)	0.4780	2940
	7		3	(8, 12)	0.6915	5625
Fold 4	6		3	(32, 12)	0.1911	1050
	5		4	(12, 11)	0.4739	3024
	6		3	(8, 10)	0.7539	6000
Fold 5	7		4	(47, 9)	0.1750	1056
	7		3	(12, 12)	0.4822	3200
	7		4	(9, 10)	0.7102	5940

Table 11: Chosen Pareto-optimal parameters for LBP on each of five cross validation folds.

Lacunarity	Parameter Set	Score	
		Window Sizes	Sep. Score
Fold 1		-0.0677	83
Fold 2		-0.0521	104
Fold 3		-0.0452	90
Fold 4		-0.0635	92
Fold 5		-0.0581	98

Table 12: Chosen Pareto-optimal parameters for Lacunarity on each of five cross validation folds. Since the set of window sizes is a list of around 100 integers between 2 and 200, we had difficulty condensing it in a presentable size. Thus, window sizes are encoded as a QR code, enabling results to be copied/pasted. Scanning the code reveals a comma-separated list of Pareto-optimal window sizes.

Haralick	Parameter Set				Score	
	Num.	Levels	Distance	Block Size	Sep. Score	Feat. Size
Fold 1	32	12	(73, 48)	9.5533	588	
	29	12	(60, 25)	45.2462	1456	
	26	11	(66, 12)	91.7909	2912	
Fold 2	27	11	(63, 30)	75.2117	1120	
	31	9	(26, 42)	140.2478	1568	
	30	11	(31, 30)	235.8086	2156	
Fold 3	32	12	(47, 47)	24.5360	980	
	26	12	(22, 48)	32.9401	1764	
	30	10	(46, 18)	130.8751	2380	
Fold 4	32	9	(71, 56)	3.6868	504	
	32	12	(71, 37)	16.2245	756	
	32	12	(44, 37)	41.2100	1260	
Fold 5	28	12	(59, 27)	29.6199	1344	
	21	12	(46, 13)	88.0654	3360	
	16	12	(20, 16)	121.8107	5852	

Table 13: Chosen Pareto-optimal parameters for Haralick features on each of five cross validation folds.

Hu Moments	Parameter Set		Score	
	Block Size		Sep. Score	Feat. Size
Fold 1	(17, 31)		-0.0338	840
	(17, 16)		-0.0283	1596
	(9, 23)		-0.0119	2254
Fold 2	(28, 16)		0.0812	1064
	(29, 11)		0.1359	1372
	(23, 8)		0.2040	2394
Fold 3	(19, 24)		-0.0244	1001
	(13, 16)		0.0519	2128
Fold 4	(23, 19)		-0.0287	1008
	(15, 17)		-0.0247	1764
Fold 5	(23, 21)		-0.0286	945
	(14, 19)		-0.0246	1680
	(23, 8)		-0.0127	2394

Table 14: Chosen Pareto-optimal parameters for Hu moments on each of five cross validation folds.

Zernike Moments	Parameter Set		Score		
	Block Size	Order	Sep.	Score	Feat. Size
Fold 1	(66, 66)	8	179.6429	500	
	(68, 61)	12	218.1904	735	
	(63, 66)	11	219.2217	840	
Fold 2	(72, 72)	7	437.6073	300	
	(64, 64)	10	479.9312	720	
	(65, 61)	12	484.0633	980	
Fold 3	(67, 54)	9	131.7052	450	
	(77, 58)	12	186.5535	882	
	(67, 59)	14	285.8775	1152	
Fold 4	(60, 42)	3	9.3558	128	
	(48, 54)	7	116.8995	600	
	(48, 53)	12	162.9375	1470	
Fold 5	(75, 66)	6	83.2748	240	
	(75, 68)	7	117.0568	300	
	(74, 55)	12	156.4977	882	

Table 15: Chosen Pareto-optimal parameters for Zernike moments on each of five cross validation folds.