SAND2019-12177J

Polarimetric SAR Image Terrain Classification

R. Derek West, Thomas E. LaBruyere III, Jacek Skryzalin,
Katherine M. Simonson, Ross L. Hansen, and Mark H. Van Benthem

***Abstract*—In practical applications of automated terrain classification from high-resolution polarimetric synthetic aperture radar (PolSAR) imagery, different terrain types may inherently contain a high level of internal variability, as when a broadly defined class (*e.g.* "trees") contains elements arising from multiple sub-classes (pine, oak, willow). In addition, real-world factors such as the time of year of a collection, the moisture content of the scene, the imaging geometry, and the radar system parameters can all increase the variability observed within each class. Such variability challenges the ability of classifiers to maintain a high level of sensitivity in recognizing diverse elements that are within-class, without sacrificing their selectivity in rejecting out-of-class elements. In an effort to gauge the degree to which classifiers respond robustly in the presence of intra-class variability and generalize to untrained scenes and conditions, we compare the performance of a suite of classifiers across six broad terrain categories from a large set of PolSAR image sets.**

The main contributions of this article are: 1) an analysis of the robustness of a variety of current state-of-the art classification algorithms to intra-class variability found in PolSAR image sets, and 2) the associated PolSAR image and feature data that Sandia is releasing to the research community with this publication. The analysis of the classification algorithms we provide will serve as a benchmark of performance for future PolSAR terrain classification algorithm research and development enabled by the image sets and data provided. By sharing our analysis and high-resolution fully-polarimetric Sandia data with the research community, we enable others to develop and assess a new generation of robust terrain classification algorithms for PolSAR.

***Keywords*—Polarimetric SAR (PolSAR), Machine Learning, Wishart, Terrain Classification**

I. Introduction

In the fields of automated classification and target recognition, a tension always exists between the *sensitivity* of an algorithm, measure, or heuristic (its ability to identify elements belonging to a designated "target" class), and its *selectivity* (ability to reject elements not belonging to the target class). Classifiers that display a high degree of selectivity may struggle to perform robustly for broadly defined target classes that show significant levels of intra-class variability. The very properties that keep false alarm rates low can translate to poor recognition of target elements whose statistical properties are slightly different from those included in the data set used for classifier training. In this paper, we will explore how various established classifiers perform in a difficult, real-world classification problem, featuring six general terrain classes that demonstrate significant between-class overlap along with high variability within classes. The challenge problem lies in the area of terrain classification for fully polarimetric synthetic aperture radar (PolSAR) imagery.

PolSAR is an active remote sensing imaging modality that measures the response of a physical scene to polarized electromagnetic waves. Complex-valued PolSAR images can be decomposed to extract the basic scattering physics associated with elements of the scene. In the vast majority of previously published work on terrain classification in the PolSAR realm, the training and testing of the classifiers is performed on different pixels from the same (single) image. While this approach enables a high degree of selectivity for narrowly-defined terrain classes, algorithms developed in this manner may show low sensitivity when tested against different Earth scenes, collected under diverse weather conditions and imaging geometries.

In contrast to earlier work, in this paper we apply and compare a suite of classifiers on data from a large and diversified set of high-resolution PolSAR imagery, containing separate training and test data from different days and multiple physical locations. As a result, the intra-class variability in our image sets is much larger than that seen in previous studies. We define broad classes of terrain, which may be multi-modal: for example, pavement can either be smooth or cracked, tree may differ in species, foliage, and moisture content. We expect that classifiers trained on such broadly defined and highly variable classes should generalize well to unseen test data, as long as there are no systematic and significant differences between the training and test data sets. However, in practical applications where classification is required for previously unseen locations, some level of training set bias is likely to exist, and classifiers that are robust to untrained conditions, without sacrificing selectivity, are most desired.

To address the variety of research interest in classification paradigms and to set a baseline for future research in PolSAR terrain classification algorithms, the set of classifiers to be studied in this paper includes methods based on physical, statistical, and data-driven models and also includes one-class, binary-class, and multi-class techniques. The performances of the various classifiers (sensitivity and selectivity) are compared on training and testing image sets, in order to determine which classifiers best demonstrate robustness to intra-class variability and generalize well to unseen test data. The image sets used in this paper were collected in central New Mexico, USA, with the FARAD PhoeniX X-band radar system, developed by Sandia National Laboratories, and flown on a de Havilland Twin Otter aircraft. Upon publication of this paper, Sandia will release a diverse set of PolSAR images representative of the data analyzed in this study. By providing these high-resolution image sets and sharing our preliminary analysis, we hope to enable research and development leading to a next generation of high-performing PolSAR terrain classification algorithms.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

II. Related Work

It is not uncommon for researchers to apply machine learning algorithms to analyze remote sensing images. Although the images analyzed are typically overhead images of land, the specific classification tasks vary (e.g., forest type classification [1], plant growth stage classification [2], etc.). However, there is a historical correlation between the remote sensing technique used to produce the image and the machine learning algorithms chosen for analysis. Whereas many researchers use mainstream machine learning and computer vision algorithms to analyze passive remote sensing images, some researchers exploit the physics behind active remote sensing systems when developing image recognition techniques.

Because the complex-valued coherency matrix computed from a homogeneous region in PolSAR data follows a complex Wishart distribution, many researchers have used the maximum likelihood estimation based Wishart classifier, for each of a specified number of training classes [1], [2], [3], [4], [5], [6]. However, the requirement to fit the Wishart distribution model requires that terrain categories be relatively homogeneous, an assumption that is seldom valid over wide geographic areas and different times of the year. For example, although a Wishart classifier might be appropriate for distinguishing between various specific crop types [2], it is less appropriate for the task of distinguishing between broad cropland and forest categories. Indeed, Lee et al. found it necessary to divide their "ocean" class into two separate classes to account for the different behaviors of the ocean surface in coastal settings [4]. Because the Wishart classifier is typically applied to individual pixels, it is possible to use a model of terrain uniformity to boost performance of the Wishart classifier by minimizing the effects of speckle. To this end, Costa Freitas et al. use a Potts-Strauss model (a Markov random field for classification on a grid) to boost performance [6].

Researchers have also applied more traditional machine learning algorithms to PolSAR classification. Various authors have augmented the variables used during Wishart classification with features measuring texture (contrast, uniformity, etc.) or features derived from a false-colorization of the PolSAR image; these features are then fed into a support vector machine (SVM) or random forest [1], [7]. Moriyama et al. construct a decision tree for terrain classification based off the polarization correlation coefficients [8]. Additionally, many authors choose to apply feedforward neural networks to PolSAR data [9], [10], [11], [12], although these models tend to be simpler than the large, complex models used in modern computer vision applications.

Our research is inspired not only by traditional PolSAR classification studies, but also by research in hyperspectral image classification, which tends to lie closer to mainstream computer vision and machine learning. The most common hyperspectral image classification algorithms include SVMs [13], [14], [15], [16], [17], [18], random forests [19], and convolutional neural networks (CNNs) [13], [20]. As with PolSAR images, some researchers studying hyperspectral images utilize texture information [14], [16], and others regularize image output using Markov random fields [18].

With both PolSAR and hyperspectral images, researchers are challenged by a relatively small amount of labeled data, and classification performance in these domains tend to suffer when moving from a supervised to an unsupervised setting [9]. Of the approaches to hyperspectral imaging using SVMs, many are semi-supervised [14], [16], [17]; these begin by training the classifier on labeled data and iteratively improve the model either by requesting labels for a small number of data instances from the user, or by slowly incorporating unlabeled data whose classifier-estimated labels are determined with high confidence. Finally, Romero et al. consider a hybrid model consisting of both a CNN and a SVM [13]. They train a CNN using a completely unsupervised technique designed to promote sparse feature representations for small training sets; this CNN is then used to extract meaningful features for input into an SVM.

III. Data

To properly train and test the classifiers for consideration in this paper, we selected data from fully-polarimetric image sets collected over a variety of scenes and scene content. The grazing angles of the collected data range from around 33° to 35°, and with the exception of one of the image sets, which had approximately a fifteen degree squint off of broadside, the collected data had relatively constant broadside imaging geometry. All of the image sets have approximately four-inch resolution in both range and azimuth. The detailed information of each image set is included in the Appendix.

The fully-polarimetric data sets contain measurements from each polarization channel, denoted as HH, HV, VH, and VV, where H denotes a horizontal polarization, V denotes a vertical polarization, and the first letter denotes the receiving polarization port and the second letter denotes the transmitting polarization port. Data from multiple passes were collected from each scene with repeated geometry to support coherent processing of the image sets. The complex-valued images formed from the measured data are denoted as I^i_{HH}, I^i_{HV}, I^i_{VH} , and I^i_{VV}, where $i \in \{1,\ 2\}$ indicates the pass it was collected. Due to the signal being transmitted and received from the same radar system, the law of reciprocity applies ($I_{HV} = I_{VH}$); thus, the cross-polarization channels were combined, $I_{CX} = \frac{1}{2}\left(I_{HV} + I_{VH}\right)$.

Further derived image products were produced from the two-pass data sets. The complex-valued I^i_{VV} images were used to form coherent change detection (CCD) [21] and Delta-C [22] images, denoted as γ and ΔC, respectively. The CCD images convey the loss of coherence between the two temporally separated passes, and the Delta-C images highlight regions where there is a loss of coherence along with a change in magnitude. The images from the first pass were processed through the G4U polarimetric decomposition [23] to produce the surface, dihedral, volumetric, and helical scattering power component maps, denoted as I_{P_s}, I_{P_d}, I_{P_v}, and I_{P_c}, respectively.

The total listing of co-registered image products processed for the classifiers in this paper are as follows: I^1_{HH}, I^1_{CX}, I^1_{VV}, I^2_{HH}, I^2_{CX}, I^2_{VV}, I_{P_s}, I_{P_d}, I_{P_v}, I_{P_c}, γ, and ΔC. All of the image products were normalized to the interval $[0\ \ 1]$; the γ and ΔC products are natively in this interval, the dynamic

range for the magnitude images was set to $[-67\ 0]$ dB and then normalized to the interval, and the G4U scattering power components were divided by the total scattering power, which normalizes them to the unit interval.

Twenty-six fully-polarimetric image sets were used for extracting model data and eighteen image sets were used for extracting blind test data. The model data were used for both training data and development (dev) data for fitting probability distributions and determining metric thresholds.

The scenes imaged contain a variety of natural and man-made objects. To aid in the determination of the robustness of classifiers, the following six broad classes of scene content, which were common to the majority of the training and test image sets, were selected for this paper: building (BLD), ground (GRD), low-return (LRT), low-vegetation (LVG), tree (TRE), and paved (PVD). The GRD class contains anything from hard-packed to loose soil regions. Similarly, the LVG class may contain anything from fields of crops to small shrubs. The BLD class is defined as the radar return from man-made structures, which can vary in the materials, structural geometries, and roof-types (pitched vs. flat). The LRT class contains measurements of regions of radar shadow and forward-scattering surfaces, both of which contain primarily system thermal noise and multiplicative noise within the scenes. These are broadly defined classes and likely contain intra-class variability. Furthermore, ambiguity may exist between some of the classes, such as hard-packed soil in the GRD class having similar radar measurements as the PVD class; similarly, some of the large shrubs in the LVG class may have similar measurements as the TRE class.

SAR images have speckle that can cause pixel-wise variability in the test and training data sets. There are methods to mitigate speckle such as speckle filters and multi-looking the data; however, these methods work best if they are applied over homogeneous regions. Image segmentation methods can be applied to identify homogeneous regions within an image. Some successful image segmentation algorithms that have been applied to PolSAR imagery include simple-linear iterative clustering (SLIC) [24], [25], spectral graph partitioning [26], and a method based on edge-maps [27].

In this paper, the SLIC algorithm was applied to group pixels within an image into so-called superpixels. The membership of pixels into superpixels is determined by a weighted combination of pixel spatial proximity and colorspace distance; thus, pixels that are close in both distances are deemed homogeneous and are formed into a superpixel. In this manner, the SLIC algorithm segmented the entirety of each image in our training and test set into relatively homogeneous superpixels. The superpixels were then multi-looked or averaged in order to reduce the speckle and are used for most of the classifiers in this paper. The following technique was employed to produce the superpixels. The I_{P_s}, I_{P_d}, I_{P_v}, I_{P_c}, and total power images were combined to form an RGB image; the I_{P_s}, I_{P_d}, and I_{P_v} images were inserted into the red, blue, and green channels respectively, the I_{P_c} image was split between the red and green channels. The resulting RGB image was transformed to the HSV color-space and the total power image was inserted into the value channel. Finally, the HSV image was transformed back to the RGB color-space. The SLIC over-segmentation algorithm was then run on the resulting RGB image, which segmented the image into superpixels.

The SLIC algorithm was set to contain approximately 200 pixels per superpixel. Since each superpixel contains pixels that are spatially proximal and close in colorspace, the borders of the superpixels nicely delineate contrasting regions of the images and the interior of each superpixel is generally a homogeneous region. The training, dev, and test data were obtained for each class by selecting superpixels from the image sets. Table I lists the number of superpixels selected for the training, dev, and test data sets.

TABLE I. SUMMARY OF THE NUMBER OF SUPERPIXELS SELECTED FOR TRAINING, DEV, AND TEST DATA FOR THE CLASSES UNDER CONSIDERATION.

	BLD	GRD	LRT	LVG	TRE	PVD
Train	1,768	4,691	6,310	7,419	8,791	1,661
Dev	196	517	700	822	975	184
Test	721	4,060	6,337	4,871	6,181	3,049

The basis for many of the classifiers in this paper are feature vectors constructed by computing the mean values of each image product, for each superpixel, for each class. To be explicit, the feature vector is of the form:

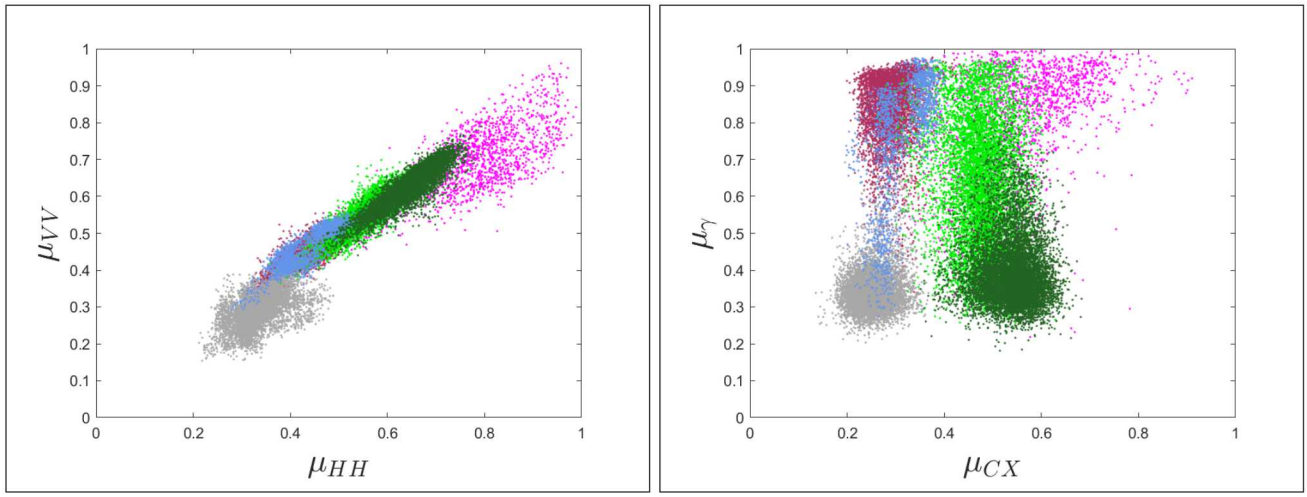
$$\mathbf{v} = \left[\mu_{P_c}\ \mu_{P_d}\ \mu_{P_s}\ \mu_{P_v}\ \mu_{\gamma}\ \mu_{\Delta C}\ \mu_{CX}\ \mu_{VV}\ \mu_{HH}\right]^T, \qquad (1)$$

where the notation μ_X indicates the mean-value of a superpixel from image product X.

Two-dimensional scatterplots can be created from pairwise combinations of the elements of the feature vectors and can help to visualize the data. A total of 36 pairwise scatterplots can be generated from the elements of the feature vectors. Figures 1(a) and 1(b) illustrate the scatterplots for two of the pairwise feature combinations. In the pairwise features illustrated, it is clear that there is a substantial amount of overlap between the GRD and PVD classes, as well as the TRE and LVG classes.

IV. CLASSIFICATION ALGORITHMS

A *classifier* is an algorithm that makes a determination about the category of a given input. In this paper, we study three kinds of classifiers: one-class classifiers, binary-class classifiers, and multi-class classifiers. Multi-class classifiers partition data into one of a set number of predetermined classes. Binary-class classifiers explicitly partition the data into class-A and class-B, which can also represent in-class and out-of-class. However, in the one-class classification paradigm, each classifier is trained to model data from one particular class; when the classifier is applied to new data, it outputs a decision on whether the data is consistent with the target class. One-class classifiers come with the benefit that training such a classifier requires only in-class data. This is especially useful when only a small handful of classes are needed, out-of-class data is difficult to obtain, or if the out-of-class data is expected to vary wildly between when the classifier is trained and when the classifier is deployed. It is easy to add or remove classes in a one-class framework without having to retrain, which is



(a) (b)

Fig. 1. Illustration of two of the possible pairwise plots of the elements of the feature vectors for (a) μ_{HH} vs. μ_{VV} and (b) μ_{CX} vs. μ_γ. The color coding represents the different classes of data: BLD - magenta, GRD - maroon, LRT - gray, LVG - light green, PVD - blue, and TRE - dark green. Notice there is a non-trivial amount of class overlap between the GRD and PVD classes, as well as the TRE and LVG classes.

a clear benefit over binary-class and multi-class classifiers. However, one-class classifiers are sometimes outperformed by their binary-class and multi-class counterparts, which can use out-of-class data to learn how to recognize when an input is not consistent with the in-class data. In the end, a user must choose between one-class, binary-class, and multi-class classifiers based on practical considerations relating to the specific problem at hand. We now give brief descriptions of the classifiers that we used in our analysis.

A. Neural Networks

Neural networks (NN) are powerful multi-class models that can be summarized as "iterative logistic regression", in which outputs of one set of logistic regressions are used as inputs to another set of logistic regressions. Neural networks can be decomposed into *layers*; each layer of a neural network consists of a linear portion (e.g., a matrix-vector multiplication) and a nonlinear *activation* (e.g., an application of the hyperbolic tangent function to each coordinate of a vector). When constructing neural networks for classification, the numeric output of a neural network is often a k-dimensional vector of positive numbers that sum to 1.

To construct our neural network, we used the Keras neural network library [28]. Our network is a "dense feed-forward network" consisting of two 18-dimensional hidden layers, each with a hyperbolic tangent activation function. We use L_2-regularization (with a regularization factor of 0.01) on the matrices learned during training. Finally, we trained our network using Adadelta [29] with the learning rate set to 0.0075 for 60000 training batches, each of which consisted of 256 points from each of our 6 training classes.

B. Convolutional Neural Networks

Convolutional neural networks (CNN) are a type of neural network that take into account spatial information. CNN have been used with high success rates in areas of image recognition, image segmentation, and many others. The advantage of the CNNs over the other classifiers explored in this paper is that it can explicitly take the texture of the terrain within the images into account. The CNNs in this paper were constructed as a series of binary-class classifiers. The training set for the CNN consisted of a 32×32 pixel chip extracted around the centroid of each superpixel for each image product. Thus, one training sample is a $32 \times 32 \times 9$ tensor from the labeled superpixel training set. Each CNN binary classifier was constructed with the same architecture, which contained a total of five layers. The first two layers consisted of four 2-D convolutional layers with a kernel size of 3×3 and used the rectified linear unit (ReLU) activation function. Following the convolutional layers was a 2-D global average pooling layer. The second to last layer was a densely connected layer with eight units and used the sigmoid activation function. The decision layer was a densely connected layer with two units using a softmax activation function. In total there were 534 trainable parameters. The architecture framework used for the CNN classifier utilized the Keras [28] and Theano [30] software packages.

C. Tree-based methods

A decision tree is a structure for multi-class classification where internal nodes represent decisions based off features seen in data and leaf nodes represent classification decisions [31]. Because decision trees are relatively weak learners prone to over-fitting, it is common to instead use an ensemble of decision trees (i.e., so-called *random forests*) for classification. Each leaf node in a decision tree records the categorical distribution of classes of data points which have been assigned to that leaf node. These probabilities are averaged and aggregated across the decision tree to provide a vector of class probabilities for each point.

We use two types of multi-class random forests provided by scikit-learn [32] for our analysis. The first, Random Forest Classifier (RFC), is a default implementation of random forests. The second, Gradient Boosting Classifier (GBC), trains the model in a forward stage-wise fashion and weighs the constituent decision trees to minimize a loss function [33]. For both variants, we used 100 decision trees and the default settings for all other parameters.

An isolation tree is a randomly constructed decision tree for one-class classification. Each internal node in an isolation tree splits the data into two using a randomly chosen threshold of a randomly chosen feature, and each leaf node is labeled with the number of data points along that branch. An isolation forest is a collection of independently constructed isolation trees [34]. An isolation forest (IF) produces, for each point, a score representing the average (over all isolation trees) of the lengths of paths leading to the given point. In general, the length of a path in an isolation tree, which leads to an outlier, is expected to be shorter than that of a path which leads to a point in a high-density region of the data. Thus, isolation forests assign lower scores to more anomalous points. For our experiments, we trained an isolation forest for each data class using the default implementation in scikit-learn [32].

D. Support Vector Machines

A support vector machine (SVM) is a supervised learning model that classifies data into one of two classes by finding the hyperplane that best separates samples from the two classes of data (which are frequently embedded into an infinite-dimensional space prior to classification). For multi-class classification, we used the "one-vs-rest" strategy; that is, we separately trained six binary-class SVMs designed to separate one target class from all other classes. For our experiments, we trained an SVM using the default implementation in scikit-learn [32]. To use a series of (binary) SVMs in a one-class framework for class C, we used scikit-learn's "ovr_decision_function", which is built off the signed distance between a data point and the separating hyperplane for the SVM trained to separate data from class C and data from all other classes.

In contrast, a one-class support vector machine (OCSVM) is an unsupervised model that finds a hyperplane that best separates the origin and the given (in-class) data, which have been embedded in an infinite-dimensional space so that all points lie on the unit sphere [35]. An arbitrary point is then classified as out-of-class if it lies on the same side of the hyperplane as the origin and as in-class otherwise. For our experiments, we trained an OCSVM for each class using the default implementation in scikit-learn [32].

E. Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a Bayesian approach that (separately) models data from each class using a multivariate normal distribution. QDA assumes that the probability $P(k|v)$ that datum v belongs to class k is proportional to $P(v|k)P(k)$, where $P(v|k)$ is given by the multivariate distribution fit to class k, and $P(k)$ is a prior probability that a datum belongs to class k (typically defined as the ratio of training points belonging to class k). The probabilities $P(k|v)$ can be used together in a multi-class classification setting. We used the default implementation of QDA given by scikit-learn [32] for our analysis.

F. Wishart Classifier

Polarimetric SAR measurements of homogeneous regions of natural terrain are modeled well by complex-valued normal distributions. Under this assumption, the constructed coherency matrices are distributed as complex-valued Wishart. The Wishart (WST) classifier [36] is a one-class classifier that is well-suited to fully-polarimetric SAR image data and is constructed from the coherency matrices computed from fully-polarimetric SAR image data. The coherency matrices are constructed from the so-called Pauli feature vector, which can be formed for each pixel location, and are defined as follows:

$$\mathbf{k} = \begin{bmatrix} I_{HH} + I_{VV} \\ I_{HH} - I_{VV} \\ I_{HV} + I_{VH} \end{bmatrix}. \tag{2}$$

The coherency matrices are formed from the Pauli feature vector by spatially averaging the $\mathbf{k}$ vectors in a local neighborhood:

$$T = \left\langle \mathbf{k}\mathbf{k}^H \right\rangle_{\mathcal{N}}, \tag{3}$$

where the notation $\langle\cdot\rangle_{\mathcal{N}}$ indicates a spatial average over the neighborhood defined by $\mathcal{N}$ and the superscript ^{H} indicates a conjugate transpose of the vector. For the Wishart classifiers in this paper, the neighborhood $\mathcal{N}$ is defined by the superpixels.

A coherency matrix can be constructed from training data for a given class of homogeneous regions and can be used as an exemplar coherency matrix for the Wishart classifier. The distance metric for the Wishart classifier is straightforward to derive and is given as:

$$d(T, T_m) = -\left(n \log |T_m| + n\text{Tr}\left(T_m^{-1} T\right)\right), \tag{4}$$

where T_m is the in-class exemplar coherency matrix computed from the training data for class m and T is a test coherency matrix, and n is the number of pixels within a superpixel. (In the derivation of the Wishart classifier n is dropped and the metric is negated; however, our superpixel-based implementation has a variable number of pixels within a superpixel and we keep the factor of n to have a meaningful comparison of the results and we keep the negation to have higher metric values indicate greater class consistency.)

G. Probabilistic Feature Fusion Classification

Probabilistic Feature Fusion (PFF) is a framework for one-class classification that allows the combination of features computed from data from one or more sources [37], [38]. The PFF framework can also be extended to combining the output scores from different classification algorithms to create a meta-classifier that has the potential to increase the overall classification performance of any individual classifier.

The feature vectors, defined in equation (1), were used in the construction of the PFF models for each class. For a given class, the mean-value of the in-class training feature vector set was computed and a centralized PCA was computed on the training data to de-correlate the features of the feature vector. The mean of the training data feature vectors and the de-correlating rotation matrix were saved as part of the PFF model for each class. The model parameters were applied to the development (dev) feature vector data. Let $\tilde{\mathbf{v}}$ denote the centralized and de-correlated dev feature vectors. The metric

$$d_i = |\tilde{\mathbf{v}}_i - \mu_{\tilde{\mathbf{v}}_i}|^n, \tag{5}$$

was computed, where $\tilde{\mathbf{v}}_i$ denotes the i^{th} entry of $\tilde{\mathbf{v}}$, $\mu_{\tilde{\mathbf{v}}_i}$ is the mean value of $\tilde{\mathbf{v}}_i$ computed from the dev set, and n is a parameter that can be tuned so that the values of d_i are modeled well by a well-known distribution.

We modeled the non-negative d_i values with a gamma distribution. The remainder of the PFF parameters, for each class, were the shape and scale distribution parameters for the d_i values and the selected values of n.

For a given class, we determined the in-class and out-of-class separability of the computed d_i values; those with poor separation were not used in PFF. Figure 2 illustrates the PFF model for the six fused features selected for the TRE class.

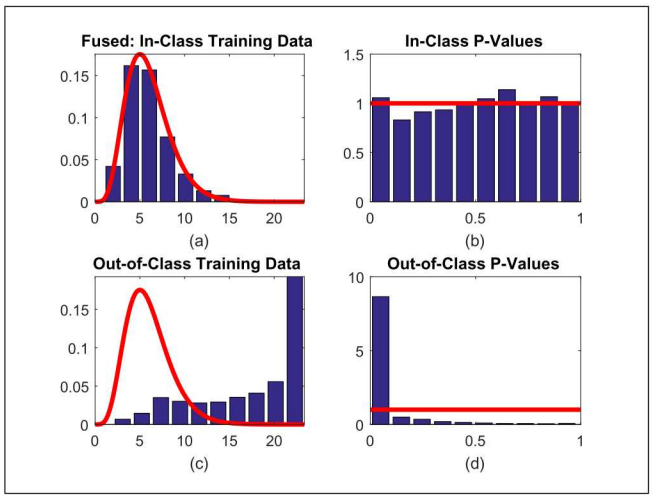


Fig. 2. Illustration of the fusion of the six selected features for the TRE class: (a) fit of the theoretic gamma distribution (red) to the histogram of the in-class fused feature metrics, (b) the theoretic uniform distribution of in-class p-values (red) to the histogram of computed p-values, (c) the separation between the theoretic in-class gamma distribution (red) and the histogram of fused out-of-class feature metrics, and (d) the theoretic in-class uniform distribution of p-values (red) to the histogram of computed out-of-class p-values (note that low p-values for out-of-class data is desirable).

V. Performance of Classifiers

With the exception of the OCSVM classifier, we summarize the results based on the metric thresholds computed from passing 90% of the dev data for each classifier. The empiric quantile values (EQVs) were computed for the OCSVM because the EQVs have the interpretation of being consistent with a class model and are more meaningful for comparisons across classes. To give a fair comparison between the classifiers in this paper, a variety of performance metrics should be considered when making comparisons and conclusions between them. We summarize results computed from receiver operating characteristic (ROC) curves, pass matrices, and confusion matrices to give an indication of the ability of a classifier to generalize to unseen data. Finally, we also created confusion images that give a visual indication of the performance of the classifiers.

A. ROC Curves

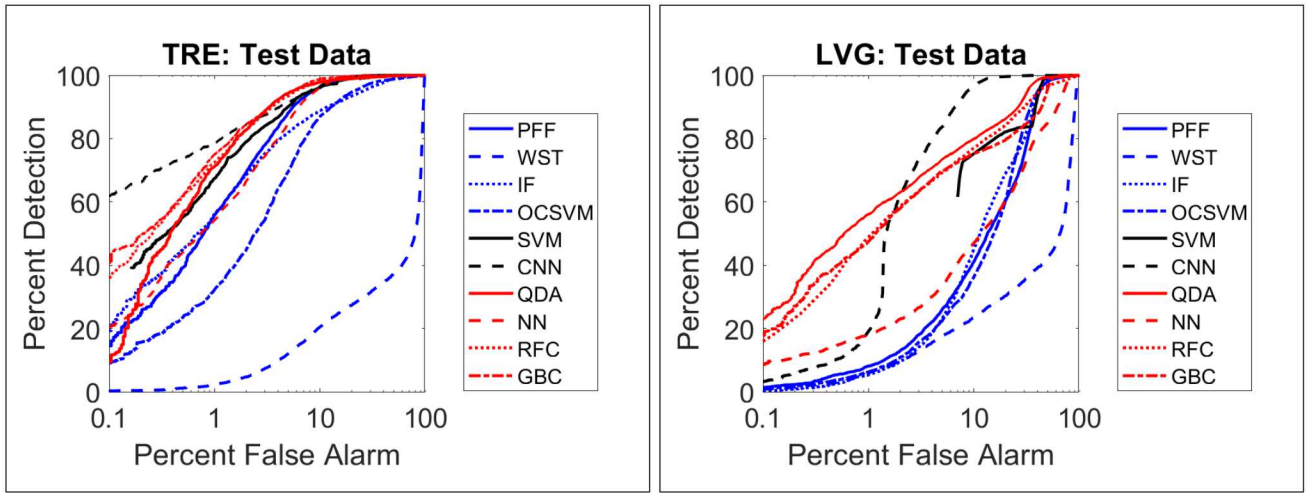
Receiver operating characteristic (ROC) curves show the level of false alarms observed for a desired level of detection, for all decision threshold levels. Furthermore, the shape of the curves give an indication of the level of sensitivity versus the selectivity of a classifier. For example, the ideal case is a classifier with a ROC curve that has a high percent detection (PD) and a correspondingly low percent false alarm (PFA) has the qualities of being selective and sensitive. The comparison of ROC curves for a given class of data, across multiple classifiers, give an indication of the relative performance between them. Figure 3(a) illustrates the ROC curves for the TRE class for the classifiers considered in this paper and figure 3(b) illustrates the same for the LVG class (note the log scaling of the x-axis).

B. Pass Matrix Analysis

A threshold can be defined for the classifier of a given class that passes a set percentage detection (PD) of the training data (*i.e.* PD = 90% to pass 90% of the dev data for a given class). For the defined classifier PD thresholds, a pass matrix summarizes the percentage of the data from all classes that pass each of the class thresholds. Ideally, the pass matrix for a classifier should have the set PD for in-class data and zero-percent for out-of-class data. As an example, the pass matrices for the training and test data from the PFF classifier are illustrated in table II with PD = 90%, determined from the dev data. Notice that 81.3% of the GRD training data (51.8% of the test data) pass the PVD threshold. There is also a large percentage of TRE data that passes as LVG, as well. An out-of-class category with a large percentage of data passing the in-class threshold doesn't necessarily indicate that incorrect decisions will be made; however, it does imply that the out-of-class category cannot be ignored as being a possible confuser to the in-class category.

Pass matrices are closely related to ROC curves. For a given operating point on a ROC curve, a pass matrix gives the PD for the operating point and breaks the PFA into the corresponding out-of-class categories. ROC curves give an overall sense of a classifiers balance between sensitivity and selectivity, whereas pass matrices further help to understand the out-of-class category sensitivities of a classifier.

Table III summarizes the in-class pass matrix performance of the various classifiers considered in this paper. The metric thresholds were found from the dev data for PD = 90%, then the thresholds were applied to the training and test data.



(a) (b)

Fig. 3. Illustration of the ROC curves computed for (a) the TRE and (b) LVG classes for the classifiers considered in this paper. Note the log-scaling of the x-axis.

TABLE II. EXAMPLE OF THE PASS MATRICES FOR THE TRAINING DATA (LEFT) AND TEST DATA (RIGHT) FROM THE PFF CLASSIFIER WITH THE METRIC THRESHOLD DETERMINED FROM THE DEV DATA WITH PD = 90%. IDEALLY, THE THRESHOLDS FROM THE DEV DATA WOULD GENERALIZE AND PASS 90% OF THE IN-CLASS TEST DATA AND THE OUT-OF-CLASS PASS PERCENTAGE WOULD BE ZERO. THE PERCENTAGES ARE ROUNDED TO A TENTH OF A PERCENT AND A '-' IS PLACED IF THE PERCENTAGES ROUNDED TO 0%.

	BLD	GRD	LRT	LVG	TRE	PVD
BLD	91.7 / 70.7	0.2 / -	- / -	8.0 / 6.7	5.7 / 4.0	0.1 / -
GRD	39.6 / 17.3	90.1 / 75.2	0.9 / -	21.4 / 7.8	- / -	81.3 / 51.8
LRT	0.3 / 0.3	0.2 / 0.1	91.3 / 87.2	0.4 / 0.1	- / -	1.3 / 3.5
LVG	29.5 / 17.9	3.6 / 11.5	0.2 / 0.1	89.3 / 63.7	42.1 / 13.3	2.7 / 16.1
TRE	15.8 / 19.9	- / -	- / 0.6	60.1 / 63.4	90.3 / 81.3	- / -
PVD	48.1 / 27.9	64.7 / 77.3	0.6 / 2.0	24.4 / 8.3	- / -	91.0 / 72.6

Ideally, the metric threshold should generalize and pass the PD level of the test data. To get an idea of the out-of-class rejection of the various classifiers, we also computed the *average* out-of-class pass percentages of the various classifiers. Table IV summarizes the percent false alarm (PFA) averaged across classes of the various classifiers. The in-class and out-of-class performance on the blind test data are also summarized in figures 4(a) and 4(b), respectively, for PD = 90%.

C. Confusion Matrix Analysis

For a given class of truth data, a confusion matrix summarizes how often the data are predicted as the correct class and incorrectly predicted as the other defined classes. Ideally, the confusion matrix for a classifier should properly predict in-class data and reject out-of-class data. As an example, the confusion matrices for the training and test data from the PFF classifier are illustrated in table V with PD = 90% set to define the unknown (UNK) category. Notice in the confusion matrices that the GRD class is predicted as PVD 27.1% of the time for the training data and 20.4% for the test data. Likewise, the TRE class is predicted as LVG a large percentage of the time.

The percent correct decision of the test data, based on a forced decision confusion matrix framework, was also used to analyze the performance of the classifiers. Table VI summarizes the percent correct decision (PCD) for the various classifiers.

D. Confusion Images

To visualize the performance of the classifiers, the classifiers were applied to superpixel data from an entire image from the training image set and the results were posterized into confusion images, where the colorization of the image indicates the predicted class. Note, because the selected image is from the *training* set, the analysis that follows does not directly address the issue of unseen test data. Although it does illustrate intra-class variability.

For a given classifier, the class decision was determined by selecting the highest metric across all classes and coloring all the pixels within the superpixel with the color associated with the selected class. If the highest metric is below the metric threshold associated with the PD = 95% threshold, then it is labeled as unknown (UNK). Figure 5 illustrates a pseudo-color G4U decomposition of an image from the training image set and a hand-truthed labeling of the image. Figures 6 and 7 show confusion images, along with accompanying binary plots that map where the confusion images differ (white pixels) from the manually-derived ground truth, for the one-class classifiers. It is also important to note that the manually-derived ground truthing, as displayed in figure 5(b), is imperfect. Figure 8

TABLE III. SUMMARY OF THE GENERALIZATION OF THE METRIC THRESHOLDS, DETERMINED FROM THE DEV DATA WITH PD = 90%, TO THE TRAINING DATA (LEFT) AND TEST DATA (RIGHT). THE HORIZONTAL LINES SEPARATE THE ONE-CLASS, BINARY-CLASS, AND MULTI-CLASS CLASSIFIERS.

	BLD	GRD	LRT	LVG	TRE	PVD
PFF	91.7 / 70.7	90.1 / 75.2	91.3 / 87.2	89.3 / 63.7	90.3 / 81.3	91.0 / 72.6
WST	93.9 / 87.8	91.9 / 95.4	90.3 / 92.9	92.1 / 93.3	90.1 / 92.9	91.6 / 96.6
IF	91.3 / 89.5	90.2 / 86.7	89.7 / 81.8	90.3 / 69.9	89.9 / 75.6	86.8 / 72.5
OCSVM	90.3 / 65.5	90.4 / 82.9	88.0 / 86.8	91.7 / 82.7	87.1 / 79.2	90.0 / 91.9
SVM	90.6 / 90.3	89.6 / 95.8	90.6 / 93.2	92.0 / 80.9	90.4 / 73.9	89.3 / 83.9
CNN	88.6 / 79.6	89.7 / 88.4	89.8 / 91.2	90.2 / 87.8	90.9 / 70.7	93.4 / 47.8
QDA	87.6 / 88.8	90.3 / 85.3	89.9 / 89.2	89.6 / 83.2	90.7 / 76.4	91.9 / 70.8
NN	89.0 / 91.7	90.1 / 95.8	89.4 / 89.5	88.9 / 65.1	90.1 / 75.3	86.2 / 89.5
GBC	92.8 / 87.1	91.2 / 72.9	91.3 / 94.3	91.4 / 76.1	92.3 / 76.4	96.0 / 51.2
RFC	100.0 / 88.1	97.5 / 72.2	95.4 / 93.2	100.0 / 79.5	100.0 / 81.2	100.0 / 59.7

TABLE IV. SUMMARY OF THE AVERAGED PFA WITH THE EMPIRIC THRESHOLD, DETERMINED FROM THE DEV DATA WITH PD = 90%, APPLIED TO BOTH THE TRAINING DATA (LEFT) AND TEST DATA (RIGHT). THE HORIZONTAL LINES SEPARATE THE ONE-CLASS, BINARY-CLASS, AND MULTI-CLASS CLASSIFIERS.

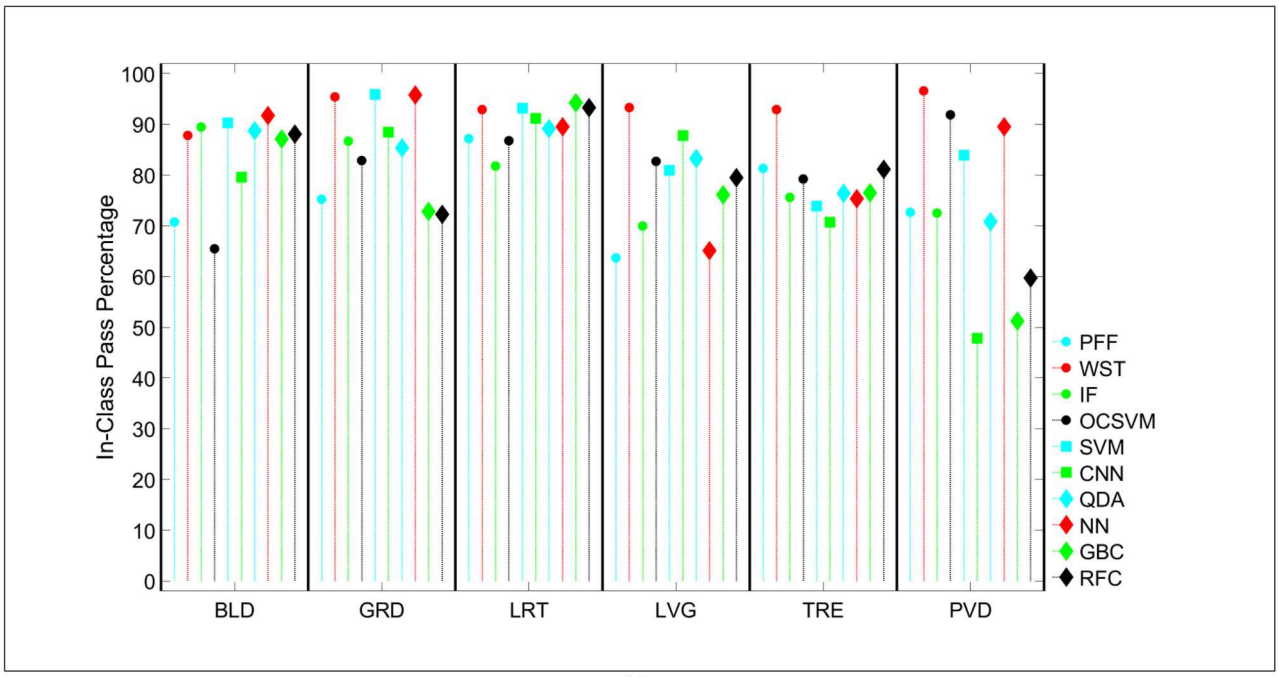
	BLD	GRD	LRT	LVG	TRE	PVD
PFF	26.7 / 16.7	13.7 / 17.8	0.4 / 0.6	22.9 / 17.2	9.6 / 3.5	17.1 / 14.3
WST	100.0 / 100.0	36.1 / 42.1	4.6 / 13.2	79.5 / 80.3	84.7 / 85.1	40.9 / 44.7
IF	4.9 / 4.7	14.3 / 18.4	0.3 / 0.4	15.7 / 15.5	7.6 / 2.7	17.7 / 14.6
OCSVM	17.0 / 15.8	17.7 / 20.2	0.7 / 1.1	25.5 / 23.6	12.7 / 5.9	25.7 / 24.0
SVM	0.8 / 1.9	13.9 / 20.9	- / 0.4	11.0 / 14.0	4.2 / 1.6	18.5 / 21.9
CNN	- / 0.1	2.1 / 15.6	- / 0.3	2.3 / 5.3	1.3 / 0.2	2.1 / 6.3
QDA	0.7 / 1.3	10.1 / 18.1	- / 0.3	8.5 / 11.3	4.5 / 1.5	11.3 / 13.5
NN	0.9 / 2.1	15.3 / 21.0	0.5 / 0.6	15.7 / 17.3	6.6 / 3.0	19.4 / 21.9
GBC	0.1 / 0.5	5.3 / 17.9	- / 0.2	5.7 / 9.9	3.1 / 1.2	6.1 / 8.6
RFC	- / 0.4	0.4 / 17.1	- / 0.2	1.7 / 10.6	0.6 / 2.0	1.9 / 11.2

TABLE V. EXAMPLE OF THE CONFUSION MATRICES FOR THE TRAINING DATA (LEFT) AND TEST DATA (RIGHT) FROM THE PFF CLASSIFIER WITH THE THRESHOLDS DETERMINED FROM THE DEV DATA WITH PD = 90%. IDEALLY, THE THRESHOLDS WOULD GENERALIZE AND PROPERLY PREDICT 90% OF THE IN-CLASS TEST DATA, THE OUT-OF-CLASS PASS PERCENTAGE WOULD BE ZERO, AND THE UNKNOWN (UNK) CLASS WOULD BE 10%. THE PERCENTAGES ARE ROUNDED TO A TENTH OF A PERCENT AND A '-' IS PLACED IF THE PERCENTAGES ROUNDED TO 0%.

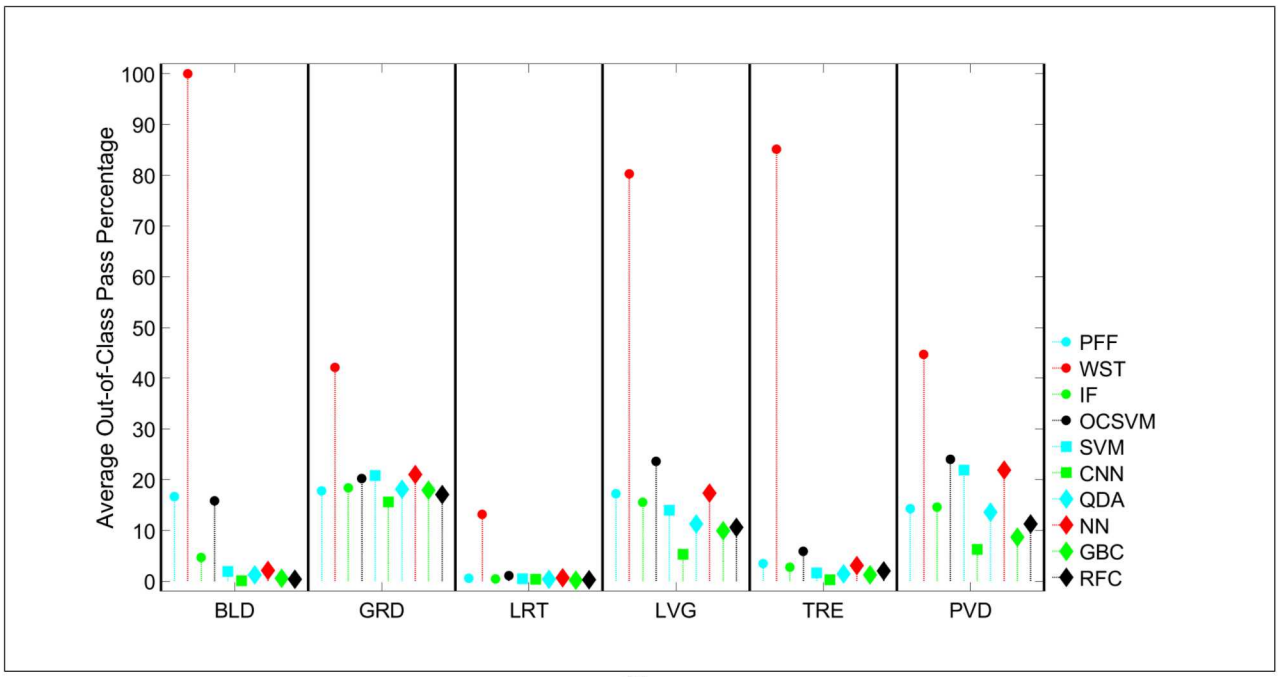
	BLD	GRD	LRT	LVG	TRE	PVD	UNK
BLD	88.1 / 69.5	- / -	- / -	2.6 / 2.9	2.7 / 1.1	- / -	6.6 / 26.5
GRD	3.2 / 2.7	62.6 / 56.3	1.0 / 0.1	3.2 / 1.6	- / -	27.1 / 20.4	3.0 / 18.9
LRT	1.3 / 0.7	0.3 / 0.2	89.7 / 85.2	0.8 / 0.1	- / -	1.5 / 4.2	6.5 / 9.5
LVG	4.7 / 4.4	1.1 / 1.6	0.2 / 0.1	75.7 / 57.1	12.2 / 3.1	0.7 / 17.2	5.4 / 16.4
TRE	3.1 / 6.4	- / -	0.2 / 0.8	12.6 / 24.6	78.4 / 60.5	- / -	5.7 / 7.8
PVD	2.1 / 1.6	19.1 / 42.6	0.4 / 3.2	3.3 / 1.5	- / -	71.0 / 39.9	4.1 / 11.2

TABLE VI. SUMMARY OF THE IN-CLASS PERCENTAGES OF CORRECT DECISIONS FROM FORCED-DECISION CONFUSION MATRICES FOR THE TRAINING DATA (LEFT) AND TEST DATA (RIGHT). THE HORIZONTAL LINES SEPARATE THE ONE-CLASS, BINARY-CLASS, AND MULTI-CLASS CLASSIFIERS.

	BLD	GRD	LRT	LVG	TRE	PVD
PFF	94.5 / 94.6	64.6 / 71.7	96.0 / 93.3	80.1 / 71.4	83.0 / 66.5	72.8 / 44.9
WST	68.7 / 68.8	77.9 / 66.9	92.5 / 95.7	80.9 / 65.9	69.4 / 48.1	49.4 / 24.5
IF	93.2 / 94.3	75.3 / 80.4	98.8 / 99.1	83.6 / 72.5	81.7 / 59.1	71.9 / 31.5
OCSVM	90.1 / 87.8	51.3 / 20.7	88.1 / 87.4	68.9 / 47.7	60.2 / 43.8	57.0 / 49.8
SVM	84.8 / 86.0	97.0 / 98.8	98.9 / 99.0	79.9 / 72.8	90.4 / 74.0	12.6 / -
CNN	99.6 / 91.5	91.3 / 86.0	99.7 / 98.3	89.9 / 77.7	95.0 / 77.1	92.2 / 38.0
QDA	90.6 / 90.8	94.0 / 90.2	97.7 / 96.3	84.9 / 80.6	82.3 / 63.6	34.2 / 3.9
NN	91.4 / 94.2	85.6 / 93.3	97.9 / 97.3	53.5 / 44.2	91.8 / 77.9	31.0 / 13.2
GBC	93.2 / 87.5	93.7 / 75.1	99.7 / 99.2	84.1 / 72.8	92.6 / 76.8	68.9 / 18.3
RFC	100.0 / 88.6	98.5 / 75.8	100.0 / 99.1	97.6 / 73.9	99.8 / 75.8	95.2 / 20.1



(a)



(b)

Fig. 4. Illustration of the pass matrix performance of the (a) in-class data and (b) the averaged percentages of the out-of-class test data that passed the in-class metric threshold; ideally, no out-of-class samples should pass the in-class threshold. The metric threshold was determined from the dev data with PD = 90%. The solid dot markers indicate the one-class classifiers, the square-shaped markers indicate the binary-class classifiers, and the diamond-shaped markers indicate the multi-class classifiers.

illustrates the same for the binary-class classifiers and figures 9 and 10 illustrate the multi-class classifiers.

There are 85,258 superpixels in the reference image; none of the superpixels in the image are from the BLD and PVD classes. We generated confusion matrices from the truth and predicted results for each classifier. Table VII summarizes the percentage of correct predictions for each of the classes for each of the classifiers. Some of the noteworthy mis-classifications in the confusion matrices include: the GRD class was predicted as both PVD and LVG for many of the classifiers; TRE was predicted as LVG by many of the classifiers (for most of the classifiers TRE was labeled as LVG more than the converse); a large number of the mis-classifications for the LRT class fell into the UNK category for many of the classifiers.

VI. Discussion of Results

Some of the classes in this data set proved to be relatively easy for the classifiers to separate; however, other classes proved to be particularly difficult. From the scatterplots in figure 1 the LRT and BLD appear to be the most separable classes. Also from the scatterplots, the TRE and LVG classes overlap one another, as do the GRD and PVD classes. It is also apparent from the scatterplot in figure 1(b) that the PVD class is multi-modal. In fact the PVD class may have proven to be the most difficult class, as is evident from the poor percentage of correct detection on the blind test data in table VI of almost all of the classifiers.

From the pass matrices in tables III and IV and the percentages of correct decisions in table VI, all of the classifiers generalize very well to the blind test data for the BLD and LRT classes. The notable exception is the performance of the WST classifier for some of the performance metrics, which will be discussed below.

In general the results from the confusion images are favorable for most classifiers. We note that the inclusion of higher-order logic could improve the performance of all classifiers on the confusion images, by taking into account the spatial proximity and predicted classes of neighboring superpixels. The convolutional neural-network effectively did employ such higher-level logic, leading to significantly smoother confusion images than those produced by the other classifiers. Developing such higher-order methds for superpixel-based classifiers could be a promising future research direction. A close comparison between the reference image and the hand-truthed image reveals that there are discrepancies between the two, and that there are regions in the reference image where the classes are ambiguous. Five of these such regions are labeled 1-5 in the hand-truthed image in figure 5(b) and are discussed below:

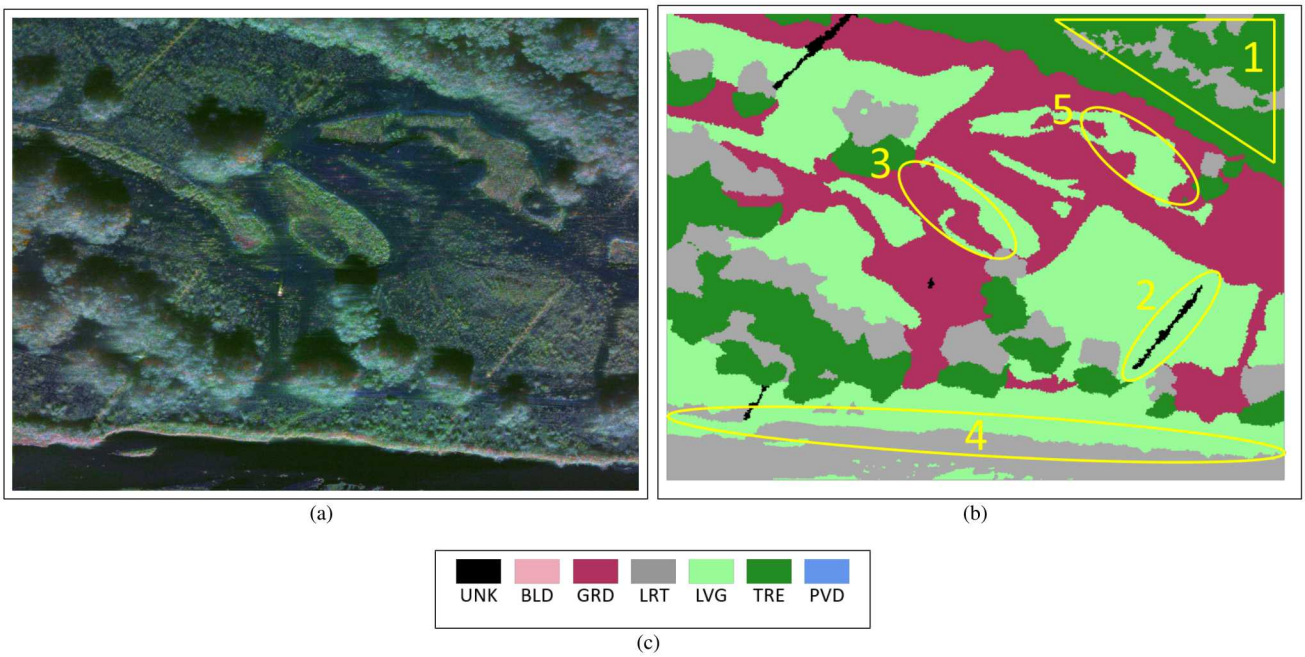
1) This area contains trees and the radar shadow due to the trees. Most of the classifiers were able to correctly label most of the tree superpixels; however, there were very few classifiers that properly labeled the low-return areas. The labels assigned to the low-return region by various classifiers included building, ground, paved, and unknown. There is a region in the transition from trees to radar shadow that contains a mixture of tree measurements, radar shadow, and multiplicative noise (due to the trees being bright reflectors followed by low-return). The classifiers may not have been trained on these regions, and thus give varied responses. In fact most of the trees in the image have some mis-labeling in these transition regions.
2) The items in the scene that are hand-truthed as unknown, are actually large metal structures to catch debris when the river overflows (there is another such structure in the upper-left region of the image, too). The response to this region varies between the classifiers as building, tree, and some unknown (from the CNN classifier). Again, the classifiers may not have been trained on measurements of this type and it is interesting that many of the classifiers are finding them to be consistent with the building class.
3) Most of this region is labeled as ground in the hand-truthed image. Comparing this area with the G4U reference image, the region is primarily ground mixed with some low-vegetation in the region. However, from the reference image, it also appears that the image is not properly focused in that region (probably due to the low-vegetation being blown by the wind). The majority of the classifiers labeled this region primarily as the low-vegetation class (with some tree class, as well), which is likely due to the Doppler energy of the unfocused response coming from the surrounding vegetation.
4) There is a row of shrubs lining the edge of the river. The shrubs are labeled as low-vegetation in the hand-truthed image, but are likely more consistent with the tree class. All of the classifiers labeled this region primarily as tree (with some building and unknown, as well).
5) This is another problematic region that could be labeled as either low-vegetation or tree. The vegetation in this region was hand-truthed to be low-vegetation, but most of the classifiers have labeled it as primarily tree. The WST classifier is the only classifier to label the region primarily as low-vegetation.

It is interesting to see that many of the mis-classifications illustrated in the binary images are due to transitions from one class to another (this is most evident in the binary mis-classification image from the CNN classifier in figure 8(d)). The polarimetric decompositions, such as the G4U decomposition that was used to generate some of the elements of the feature vector, use a spatial average to compute the decomposition parameters. The spatial average introduces a mixture of classes at the boundaries of the classes. The width of these mis-classification regions could be reduced by utilizing a smaller neighborhood when estimating the decomposition parameters, however, this would come at the risk of higher variance in the estimated parameters.

The choice of whether to use a one-class, binary-class, or multi-class classifier depends on the problem at hand and a careful consideration of the data. Further refining the selection of which classifier to use within a classifier paradigm depends

TABLE VII. SUMMARY OF THE PERCENTAGE OF CORRECT PREDICTIONS OF THE CONFUSION IMAGE FROM THE VARIOUS CLASSIFIERS FOR THE CLASSES CONTAINED WITHIN THE REFERENCE IMAGE WITH THE THRESHOLD DETERMINED FROM THE DEV DATA WITH PD = 95%. THE TOTAL PERCENTAGE OF CORRECT PREDICTIONS IS ALSO GIVEN. THE DOUBLE VERTICAL LINES SEPARATE THE ONE-CLASS, BINARY-CLASS, AND MULTI-CLASS CLASSIFIERS.

	PFF	WST	IF	OCSVM	SVM	CNN	QDA	NN	GBC	RFC
GRD	31.7	49.4	31.8	24.6	57.9	49.1	53.5	36.8	52.2	51.3
LRT	73.9	62.5	70.5	57.2	74.2	65.5	73.9	65.7	70.0	68.2
LVG	59.5	63.6	65.7	44.4	66.2	68.6	68.2	42.7	68.6	69.5
TRE	61.3	38.9	55.5	44.1	74.0	82.0	64.9	72.8	77.2	78.4
UNK	6.9	5.0	5.4	3.8	1.2	16.3	-	4.3	0.5	0.2
Total	55.5	54.5	55.8	41.8	67.0	65.6	64.6	51.8	66.2	66.2



(a) (b) (c)

Fig. 5. Illustration of the (a) G4U reference image and (b) the hand-truthed image of a scene from the set of training images. The reference image does not contain the PVD or BLD classes. (Note, the hand-truthed image may contain errors in the labels.)

on the desired balance between sensitivity and selectivity. The goal of this paper is to explore the robustness and generalization properties of various classifiers within each classifier paradigm.

A. One-Class Classifiers

It can be seen in the stem-plots in figures 4(a) and 4(b) that the one-class classifiers did fairly well on the pass-matrix metrics. According the the results in table VI, the one-class classifiers were out-performed on many of the classes. However, it is interesting to see that they did generalize to the PVD blind test data better than most of the binary- and multi-class classifiers.

1) Probabilistic Feature Fusion Classifier: The PFF classifier had very consistent performance. The PFF lacked in-class sensitivity for many of the classes, as shown in table III. However, with respect to the other one-class-classifiers it does do well with selectivity, as is evident in table IV and its relative percent correct decisions compared to the other one-class classifiers in table VI. An attribute that the PFF classifier has, is that if an incorrect decision is made, the decision can be traced back to the defined features.

2) Wishart Classifier: The WST classifier demonstrated high in-class sensitivity, as shown in table III; however, the results in table IV and figure 4 show that it generally lacks selectivity. The WST classifier is based on the distribution of the coherency matrices, which are complex-valued Wishart distributed for natural clutter. Furthermore, the Wishart classifier assumes all samples (training and test) are independent and identically distributed from the same *unimodal* statistical distribution. When data within a class come from discrete sub-classes, the assumptions underlying the Wishart model are violated; the method is not robust to such violations. When the model is valid, the Wishart classifier is able to perform very well in discriminating out-of-class. To illustrate this, consider the LRT class in figures 4 (a) and (b). Aside from some possible multiplicative noise, the LRT class should be relatively unimodal. For this class, the Wishart classifier does well in both sensititivity and selectivity across the image sets. Thus, we attribute the challenges the WST had with the other classes considered to intra-class variability. To illustrate

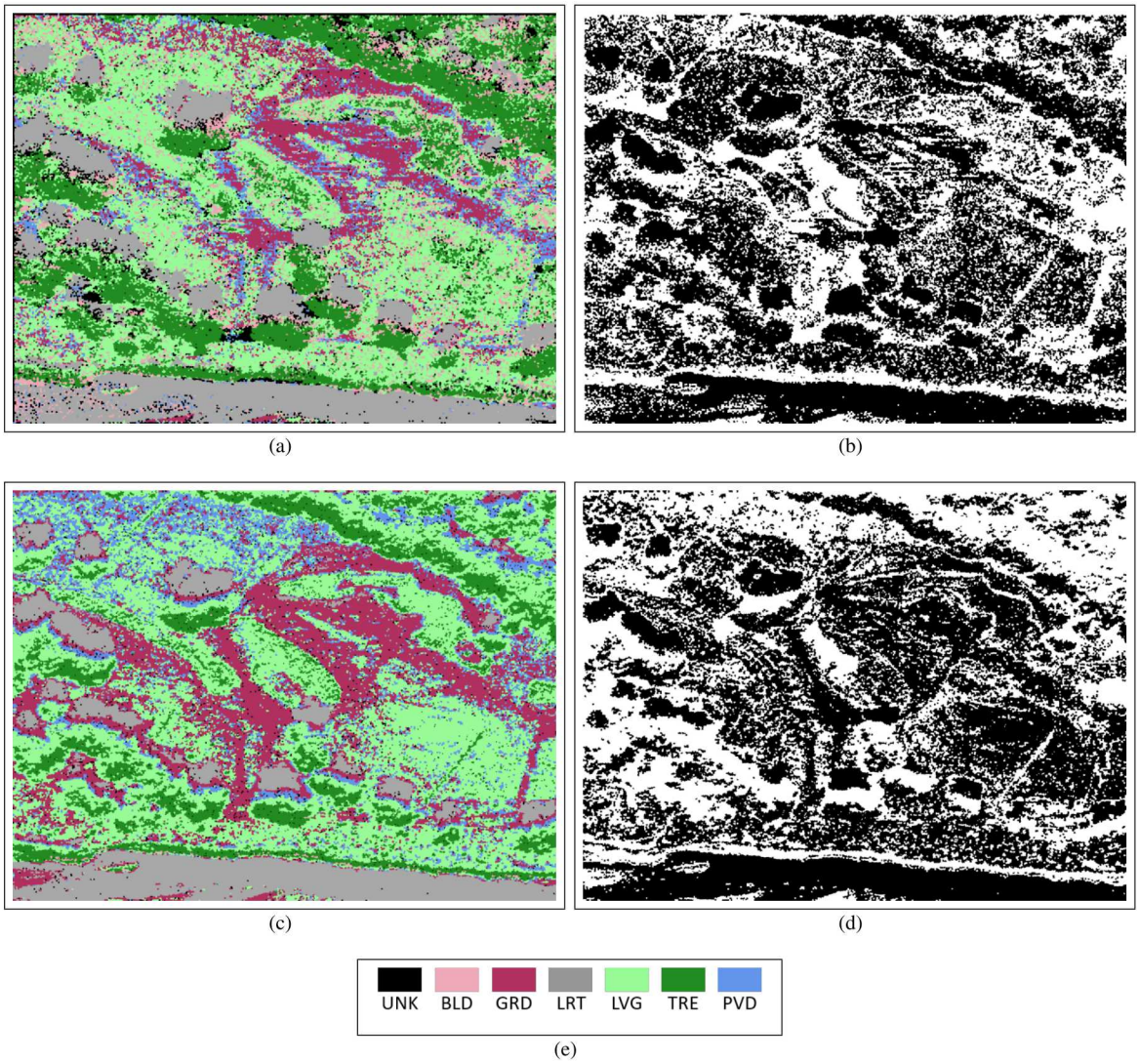


Fig. 6. Illustration of the confusion images formed from the one-class classifiers: (a) PFF and (c) WST and the associated images that illustrate where the predictions differ from the hand-truthed image (white pixels), (b) and (d), respectively.

the intra-class variability, consider the bi-modal histogram of the first entry of the coherency matrix, $T_{PVD}(1,1)$, from the PVD class training data in figure 11. According to theory the diagonal elements of the coherency matrices should have gamma distributions, which can be seen in figure 11 is not the case, confirming that the complex-Wishart model assumption cannot be made for the training data aggregated from multiple image sets.

To combat the intra-class variability issue, a class may be broken into sub-classes and sub-class Wishart classifiers can be trained. Sub-dividing a class may work if the training data contains all variations of a class that the classifier may encounter in the test data; however, if not all the variability is captured, then the classifiers may still not generalize well to unseen test data.

3) Isolation Forest Classifier: The IF classifier performance is arguably one of the better of the one-class classifiers. It demonstrated in-class sensitivity and selectivity. The main challenge the IF classifier had, was the percent correct detections with the PVD class and to a lesser degree with the TRE class. These two classes are likely to contain the most intra-class variability, which indicates that the IF classifier may be particularly challenged by this condition.

4) One-Class Support Vector Machine Classifier: The OCSVM struggled with both in-class sensitivity and selectivity. From the results in tables IV and VI, it is evident that the OCSVM is skewed more toward selectivity in the sensitivity vs. selectivity balance. However, it is noteworthy to point out that of all the classifiers, the OCSVM generalized the best to the blind test data for the PVD class in the percentage of correct decisions.

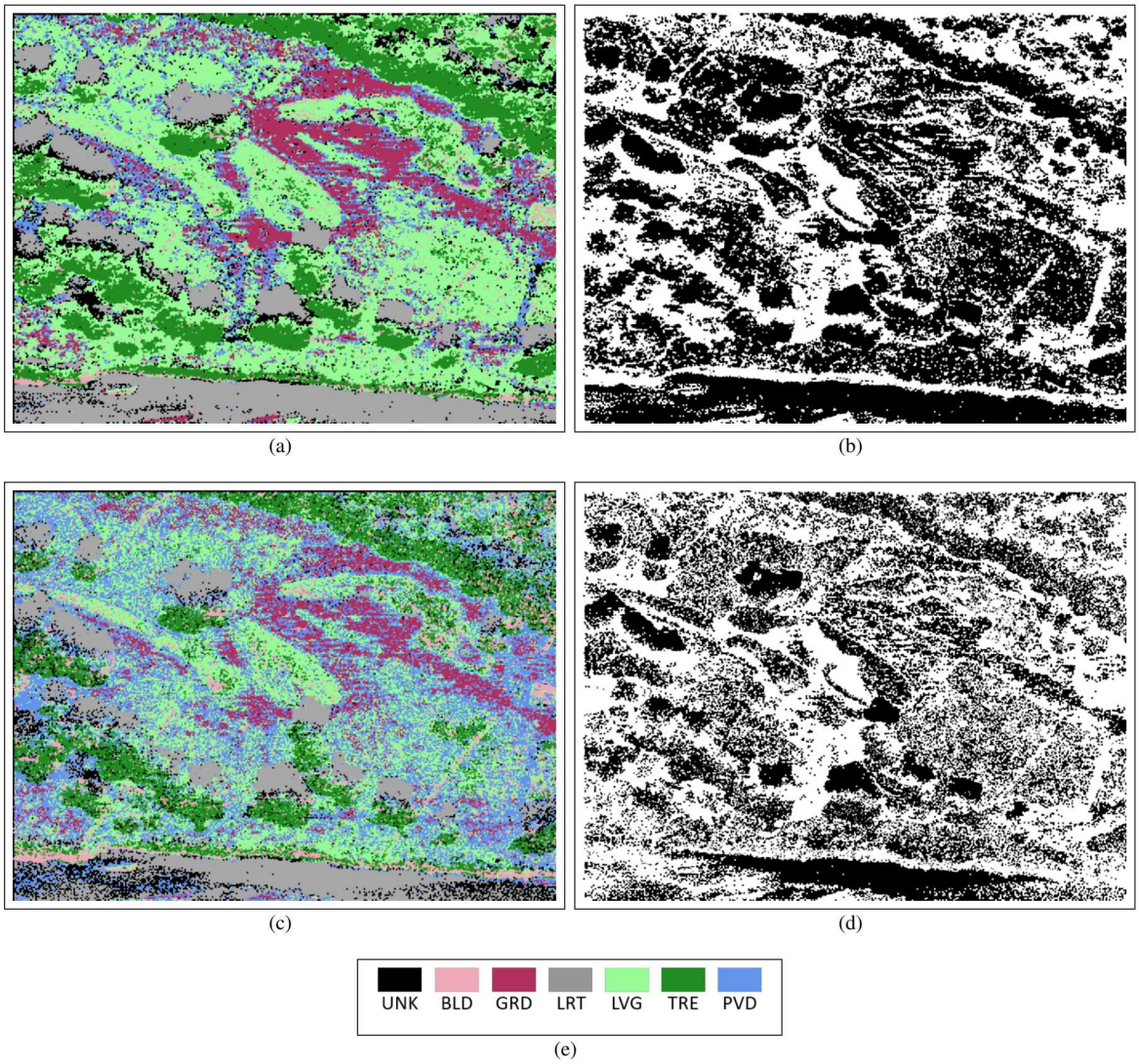


Fig. 7. Illustration of the confusion images formed from the one-class classifiers: (a) IF and (c) OCSVM and the associated images that illustrate where the predictions differ from the hand-truthed image (white pixels), (b) and (d), respectively.

B. Binary-Class Classifiers

Overall, the binary-class classifiers had excellent performance. Between the two binary-class classifiers considered in this paper, the SVM demonstrated more in-class sensitivity and the CNN showed more selectivity. The main challenge for these classifiers, was generalizing to the PVD class, as can be seen in table VI.

1) Support Vector Machine Classifier: In general, the SVM had good performance. The SVM is sensitive to the in-class data, as evident from tables III and VI and demonstrated selectivity, as shown in table IV. It is also important to note that the SVM predicted the highest percentage of correct decisions in the example confusion image, as given in table VII. The main challenge for the SVM was the percentage of correct decisions for the PVD class, where it had the lowest metrics for both training and test data of all the classifiers.

2) Convolutional Neural-Network Classifier: The CNN classifier is arguably one of the best classifiers considered in this paper. With the exception of generalizing from the training to test data for the PVD class, the CNN classifier had excellent measures in tables III, IV, and VI. Because it was trained differently, the CNN classifier is also the only classifier that was able to discriminate based on the texture in the PolSAR image sets. It is also interesting to note that the CNN classifier produced the most homogeneous regions in the confusion image of all the classifiers, as illustrated in figure 8(c), which again is likely due to how the CNN was trained.

C. Multi-Class Classifiers

Like the one-class and binary-class classifiers, the PVD class proved to be a challenge for the multi-class classifiers; otherwise, the multi-class classifiers had excellent performance

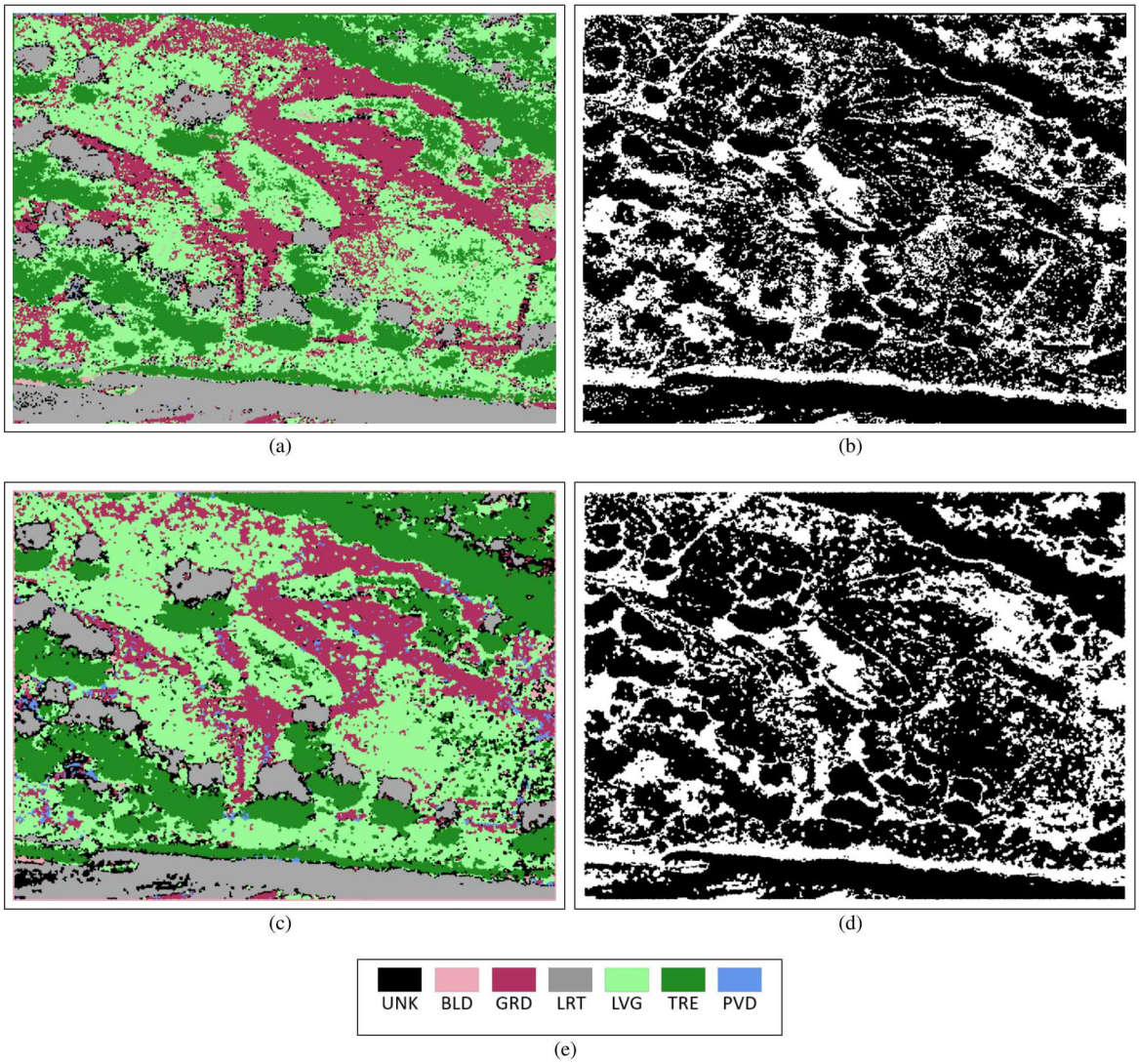


Fig. 8. Illustration of the confusion images formed from the binary-class classifiers: (a) SVM and (c) CNN and the associated images that illustrate where the predictions differ from the hand-truthed image (white pixels), (b) and (d), respectively.

measures. The decision boundaries established in training the multi-class classifiers may be too rigid for classes subject to intra-class variability when the classes have a significant amount of overlap; this could be further compounded when the training data is not fully representative of what may be in the blind test data.

1) Quadratic-Discriminant Analysis Classifier: Relative to the other multi-class classifiers, the QDA classifier is skewed more toward in-class sensitivity, as is evident in figures 4(a) and 4(b). Compared across all the classifiers considered, the QDA classifier is also one of the more selective classifiers. The main difficulty for the QDA classifier was PVD class.

2) Neural-Network Classifier: Of the multi-class classifiers, the NN classifier likely has the most in-class sensitivity as is evident in tables III and VI, but also seemed to be the least selective as can be seen in table IV. The NN classifier had difficulty with the challenging LVG and PVD classes. Furthermore, the NN classifier incorrectly predicted many superpixels as PVD in figure 9(c).

3) Gradient-Boosted Classifier: Overall, the performance of the GBC was very good; it showed a good balance between in-class sensitivity and selectivity, with skew towards selectivity, as evident in analyzing tables III and IV. Relative to all the other classifiers, the GBC classifier did very well in predicting the correct classes in figures 10(c). The main challenge for the GBC classifier was the PVD class, where it didn't do so well generalizing from the training to the test data in table III or correctly predicting the PVD class in table VI.

4) Random Forest Classifier: The performance of the RFC is very similar to the GBC. This is somewhat expected due to the fact that the GBC is also a tree-based classifier with a different training optimization metric. However, the RFC does

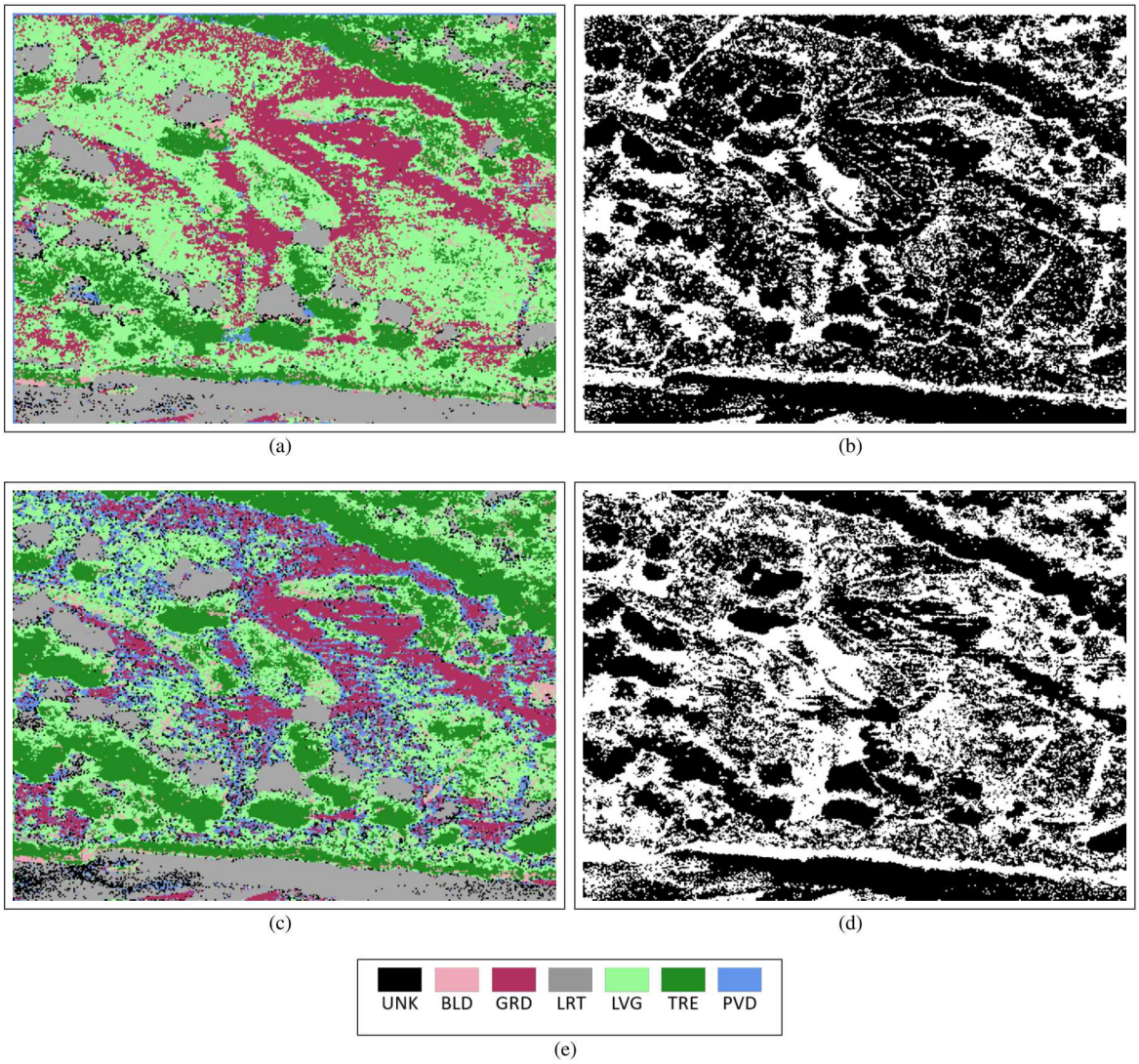


Fig. 9. Illustration of the confusion images formed from the multi-class classifiers: (a) QDA and (c) NN and the associated images that illustrate where the predictions differ from the hand-truthed image (white pixels), (b) and (d), respectively.

seem slightly more skewed towards in-class sensitivity as can be determined from figures 4(a) and 4(b) and table VI. The RFC classifier predicted about the same total percentage of correct decisions as the GBC classifier in the confusion image illustrated in figure 10(a).

VII. Conclusion

In this paper we have explored a variety of state-of-the-art one-class, binary-class, and multi-class classifiers with the goal of determining which classifiers generalize well to unseen test data, which is an important attribute when determining a classifier to use in real-world applications that contain data variability.

We trained and tested the classifiers in this paper with a relatively large set of PolSAR training and test image sets, which contained in-class variability and challenging classes.

In general, most of the classifiers considered performed relatively well on the blind test data. The notable exception was the Wishart classifier, which demonstrated good sensitivity to in-class data, but poor selectivity to out-of-class data. Given the intra-class variability of both the training and test data, we attribute the poor performance of the Wishart classifier to a model mis-match between the expected distribution of the coherency matrices with the actual coherency matrices computed from the intra-class variability. On the other hand, the neural-network and convolutional neural-network demonstrated good sensitivity and selectivity, for the majority of the classes. These networks do not assume an underlying model, which may be a positive attribute to consider for robustness to unseen data, but have the challenge of ensuring enough in-class and out-of-class data are provided for proper training.

In terms of utilizing one-class, binary-class, or multi-class

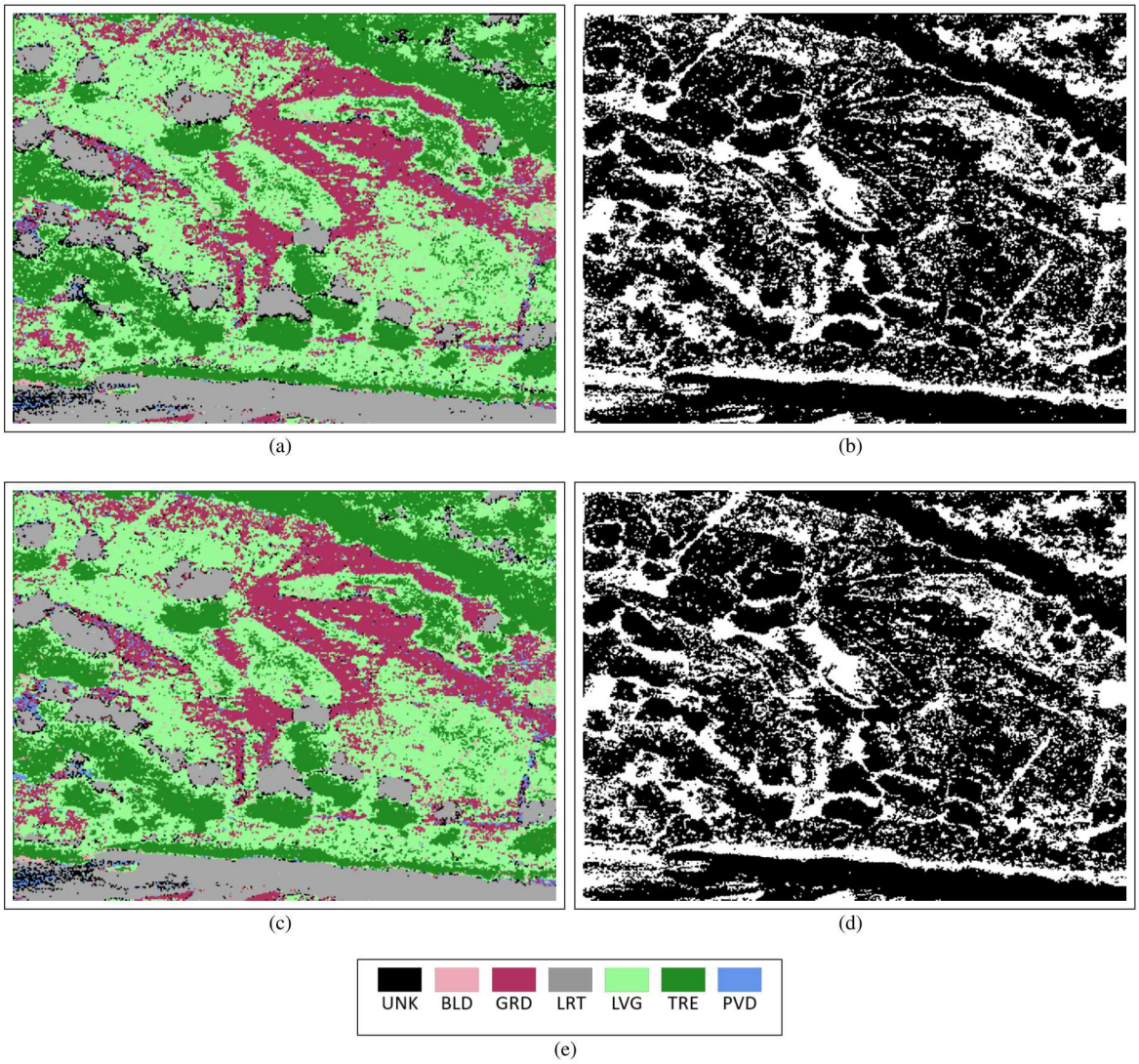


Fig. 10. Illustration of the confusion images formed from the multi-class classifiers: (a) RFC and (c) GBC and the associated images that illustrate where the predictions differ (white pixels) from the hand-truthed image, (b) and (d), respectively.

classifiers, we do not conclude or claim that one paradigm is better than another as this depends on the problem at hand; examples can be found of each type performing well. We considered the different paradigms due to research interest. One-class classifiers are useful if a single class is of interest and has unique features that naturally separate it from other classes, or if the other classes cannot be readily be modeled. An additional benefit to one-class classifiers is that new classes can be modeled without having to retrain previous classes. Binary-class classifiers can be employed if one class is desired and there is either only one other class, or if the second class is an aggregate of "don't care" classes. If enough of the "world" can be modeled and classifying all the categories is desired, the multi-class classifiers perform very well; however, retraining for all classes is required whenever additional classes are added. Time spent considering the problem at hand and understanding the data and which model fits the data will help to select the proper classification paradigm and algorithm.

Aside from the convolutional neural-network data, the superpixel-based feature-vector data used to train the various classifiers will be made available for researching the next generation of PolSAR terrain classification algorithms. Furthermore, Sandia National Laboratories will release a subset of the co-registered high-resolution image sets to facilitate further SAR/PolSAR research.

VIII. Acknowledgments

The authors would like to thank the anonymous reviewers for their thoughtful and insightful comments and suggestions that improved the content and clarity of the paper.

This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in

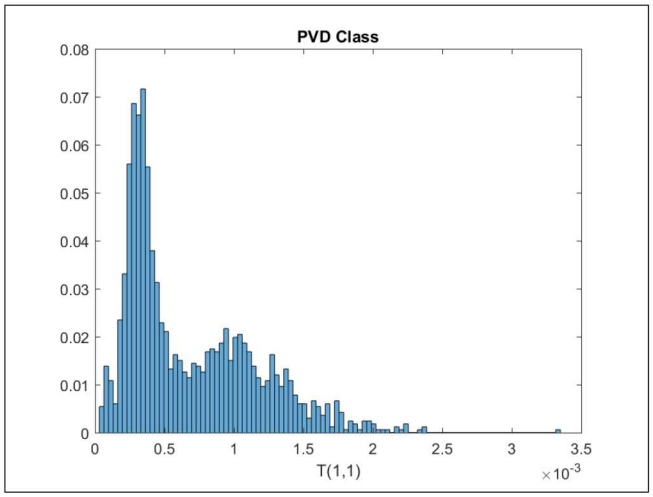


Fig. 11. Histogram of the first entries of the covariance matrices from the PVD class training data. The two modes indicate intra-class variability.

the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

This work was supported by the Laboratory Directed Research and Development program at Sandia National Laboratories. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

REFERENCES

[1] S. Middinti, C. S. Jha, and T. B. Reddy, "Forest type classification with combination of advanced polarimetric decompositions and textures of L-band synthetic aperture radar data," *Journal of Applied Remote Sensing*, vol. 11, no. 1, 2017.

[2] L. Mascolo, J. M. Lopez-Sanchez, F. Vicente-Guijalba, F. Nunziata, M. Migliaccio, and G. Mazzarella, "A complete procedure for crop phenology estimation with PolSAR data based on the complex Wishart classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 11, pp. 6505–6515, 2016.

[3] J.-S. Lee, M. R. Grunes, T. L. Ainsworth, L.-J. Du, D. L. Schuler, and S. R. Cloude, "Unsupervised classification using polarimetric decomposition and the complex Wishart classifier," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 5, pp. 2249–2258, 1999.

[4] J.-S. Lee, M. R. Grunes, and R. Kwok, "Classification of multi-look polarimetric SAR imagery based on complex Wishart distribution," *International Journal of Remote Sensing*, vol. 15, no. 11, pp. 2299–2311, 1994.

[5] L. Ken Yoong, S. C. Liew, L. K. Kwoh, and M. Nakayama, "Land cover classification using NASA/JPL polarimeric synthetic aperture radar (PolSAR) data," in *Paper presented at the 22nd Asian Conference on Remote Sensing*, vol. 5, 2001, p. 9.

[6] C. da Costa Freitas, L. de Souza Soler, S. J. S. Sant'Anna, L. V. Dutra, J. R. Dos Santos, J. C. Mura, and A. H. Correia, "Land use and land cover mapping in the Brazilian Amazon using polarimetric airborne P-band SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 10, pp. 2956–2970, 2008.

[7] S. Uhlmann and S. Kiranyaz, "Integrating color features in polarimetric SAR image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 4, pp. 2197–2216, 2014.

[8] T. Moriyama, S. Uratsuka, T. Umehara, M. Satake, A. Nadai, H. Maeno, K. Nakamura, and Y. Yamaguchi, "A study on extraction of urban areas from polarimetric synthetic aperture radar image," in *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, vol. 1. IEEE, 2004.

[9] Y. Hara, R. G. Atkins, S. H. Yueh, R. T. Shin, and J. A. Kong, "Application of neural networks to radar image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 32, no. 1, pp. 100–109, 1994.

[10] F. Del Frate, G. Schiavon, D. Solimini, M. Borgeaud, D. H. Hoekman, and M. A. Vissers, "Crop classification using multiconfiguration C-band SAR data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, no. 7, pp. 1611–1619, 2003.

[11] K.-S. Chen, W. Huang, D. Tsay, and F. Amar, "Classification of multifrequency polarimetric SAR imagery using a dynamic learning neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 34, no. 3, pp. 814–820, 1996.

[12] Y.-C. Tzeng and K.-S. Chen, "A fuzzy neural network to SAR image classification," *IEEE Transactions on Geoscience and remote Sensing*, vol. 36, no. 1, pp. 301–307, 1998.

[13] A. Romero, C. Gatta, and G. Camps-Valls, "Unsupervised deep feature extraction for remote sensing image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1349–1362, 2016.

[14] M. Marconcini, G. Camps-Valls, and L. Bruzzone, "A composite semisupervised SVM for classification of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 2, pp. 234–238, 2009.

[15] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[16] G. Camps-Valls, D. Tuia, L. Bruzzone, and J. A. Benediktsson, "Advances in hyperspectral image classification: Earth monitoring with statistical learning methods," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 45–54, 2014.

[17] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for semisupervised classification of remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 11, pp. 3363–3373, 2006.

[18] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 4, pp. 736–740, 2010.

[19] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[20] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, 2015, pp. 4959–4962.

[21] C. V. Jakowatz, Jr., D. E. Wahl, P. E. Eichel, D. C. Ghiglia, and P. A. Thompson, *Spotlight-mode Synthetic Aperture Radar: A Signal Processing Approach*. Springer, 1996.

[22] R. Sabry, "A new coherency formalism for change detection and phenomenology in SAR imagery: A field approach," *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 3, pp. 458–462, July 2009.

[23] G. Singh, Y. Yamaguchi, and S. E. Park, "General four-component scattering power decomposition with unitary transformation of coherency matrix," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 3014–3022, May 2013.

[24] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels," *EPFL Technical Report 149300*, June 2010.

[25] F. Qin, J. Guo, and F. Lang, "Superpixel segmentation for polarimetric SAR imagery using local iterative clustering," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 1, pp. 13–17, Jan 2015.

[26] K. Ersahin, I. G. Cumming, and R. K. Ward, "Segmentation and classification of polarimetric SAR data using spectral graph partitioning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 1, pp. 164–174, Jan 2010.

[27] B. Liu, H. Hu, H. Wang, K. Wang, X. Liu, and W. Yu, "Superpixel-based classification with an adaptive number of classes for polarimetric SAR images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 907–924, Feb 2013.

[28] F. Chollet *et al.*, "Keras," https://github.com/fchollet/keras, 2015.

[29] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[30] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016. [Online]. Available: http://arxiv.org/abs/1605.02688

[31] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[33] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367–378, 2002.

[34] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 413–422.

[35] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, J. C. Platt *et al.*, "Support vector method for novelty detection." in *NIPS*, vol. 12, 1999, pp. 582–588.

[36] J. Lee and E. Pottier, *Polarimetric Radar Imaging: From Basics to Applications*. CRC Press, 2002.

[37] K. M. Simonson, "Probabilistic fusion of ATR results," Sandia National Laboratories, Albuquerque, New Mexico 87185 and Livermore, California 94550, SAND Report SAND98-1699, August 1998.

[38] K. M. Simonson, R. D. West, R. L. Hansen, T. E. LaBruyere, and M. H. Van Benthem, "A statistical approach to combining multisource information in one-class classifiers," *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 10, no. 4, pp. 199–210, 2017. [Online]. Available: http://dx.doi.org/10.1002/sam.11342

IX. Appendix

The collection dates, imaging geometries, and time between repeated passes for both the training and test images sets are detailed in this appendix. The data acquisition parameters listed in tables VIII and IX are defined as follows: ψ is the grazing angle, θ is the squint angle ($+90^\circ$ defines right-looking broadside and -90° defines left-looking broadside), ϕ is the heading angle, ρ_{Az} is the azumith resolution, ρ_{Rg} is the range resolution, Range is the distance of the sensor to the scene center at mid-aperture, baseline is the mid-aperture range difference between the reference and match passes, and Δt is the time between the mid-aperture points of the reference and match passes.

TABLE VIII. SUMMARY OF THE GEOMETRY PARAMETERS AND IMAGE RESOLUTION FOR THE TRAINING IMAGE SETS. THE TOP ROW GIVES THE PARAMETERS FOR THE REFERENCE PASS IN THE IMAGE SET AND THE BOTTOM ROW GIVES THE PARAMETERS FOR THE MATCH PASS.

Image Set	Collection Date	ψ (degrees)	θ (degrees)	ϕ (degrees)	ρ_{Az} (m)	ρ_{Rg} (m)	Range (km)	Baseline (m)	Δ t (seconds)
		34.96	89.59	206.42	0.1117	0.1016	4.078		
1	07/19/2013	34.96	88.91	207.11	0.1117	0.1016	4.081	2.87	725
		34.96	89.74	249.26	0.1117	0.1016	4.133		
2	07/19/2013	35.01	89.42	249.61	0.1117	0.1016	4.126	8.58	663
		34.88	-90.53	69.54	0.1117	0.1016	4.138		
3	07/19/2013	34.97	-90.29	69.28	0.1117	0.1016	4.130	10.37	660
		35.10	-90.29	69.29	0.1117	0.1016	4.121		
4	07/19/2013	35.03	-89.19	68.19	0.1117	0.1016	4.125	6.12	661
		34.98	-90.72	69.74	0.1117	0.1016	4.131		
5	07/19/2013	35.01	-89.90	68.92	0.1117	0.1016	4.125	6.09	660
		35.00	90.04	181.98	0.1117	0.1016	4.135		
6	07/19/2013	34.98	89.09	182.92	0.1117	0.1016	4.125	10.53	640
		35.01	-89.10	25.12	0.1117	0.1016	4.077		
7	07/22/2013	35.04	-89.91	25.92	0.1117	0.1016	4.081	4.36	838
		34.98	-89.99	14.99	0.1117	0.1016	4.133		
8	07/22/2013	34.92	-89.33	14.36	0.1117	0.1016	4.141	8.39	633
		35.04	-90.97	15.99	0.1117	0.1016	4.134		
9	07/22/2013	34.92	-91.52	16.55	0.1117	0.1016	4.128	10.05	632
		34.99	-89.53	14.54	0.1117	0.1016	4.135		
10	07/22/2013	35.08	-88.69	13.71	0.1117	0.1016	4.132	6.57	633
		35.02	-89.12	14.13	0.1117	0.1016	4.133		
11	07/22/2013	34.99	-90.87	15.89	0.1117	0.1016	4.130	4.17	633
		34.97	90.06	194.95	0.1117	0.1016	4.135		
12	07/22/2013	35.02	89.56	195.45	0.1117	0.1016	4.135	3.41	677
		35.00	-89.68	14.69	0.1117	0.1016	4.134		
13	07/22/2013	35.02	-90.72	15.72	0.1117	0.1016	4.137	4.07	714
		35.01	-90.08	15.10	0.1117	0.1016	4.135		
14	07/22/2013	35.12	-88.66	13.66	0.1117	0.1016	4.129	10.08	714
		35.03	-89.84	14.86	0.1117	0.1016	4.131		
15	07/22/2013	34.95	-90.26	15.29	0.1117	0.1016	4.133	6.29	714
		34.99	-90.63	15.65	0.1117	0.1016	4.133		
16	07/22/2013	35.02	-90.04	15.06	0.1117	0.1016	4.130	3.19	714
		35.08	90.64	194.38	0.1117	0.1016	4.130		
17	07/22/2013	35.03	90.30	194.72	0.1117	0.1016	4.135	5.62	677
		34.91	-90.10	69.10	0.1117	0.1016	4.125		
18	07/24/2013	34.88	-93.11	72.11	0.1117	0.1016	4.122	3.98	759
		35.11	-88.96	67.95	0.1117	0.1016	4.124		
19	07/24/2013	35.05	-87.60	66.60	0.1117	0.1016	4.132	8.85	760
		34.92	89.53	249.50	0.1117	0.1016	4.117		
20	07/24/2013	34.99	90.45	248.58	0.1117	0.1016	4.126	10.12	753
		35.04	89.63	182.40	0.1117	0.1016	4.137		
21	07/26/2013	35.04	89.66	182.38	0.1117	0.1016	4.129	7.81	619
		34.97	89.33	182.69	0.1117	0.1016	4.135		
22	07/26/2013	34.99	90.18	181.85	0.1117	0.1016	4.133	2.33	619
		34.96	-89.03	25.05	0.1117	0.1016	4.083		
23	07/29/2013	34.99	-90.81	26.83	0.1117	0.1016	4.079	3.91	978
		34.99	89.26	206.76	0.1117	0.1016	4.082		
24	07/29/2013	34.89	88.73	207.28	0.1117	0.1016	4.088	9.52	990
		34.04	-104.71	25.74	0.1116	0.1016	4.176		
25	07/29/2013	34.07	-104.83	25.85	0.1116	0.1016	4.176	1.89	310
		33.14	-91.43	3.48	0.1015	0.1016	3.976		
26	04/01/2015	33.48	91.55	180.47	0.1015	0.1016	3.991	27.51	201

TABLE IX. SUMMARY OF THE GEOMETRY PARAMETERS AND IMAGE RESOLUTION FOR THE TEST IMAGE SETS. THE TOP ROW GIVES THE PARAMETERS FOR THE REFERENCE PASS IN THE IMAGE SET AND THE BOTTOM ROW GIVES THE PARAMETERS FOR THE MATCH PASS.

Image Set	Collection Date	ψ (degrees)	θ (degrees)	ϕ (degrees)	ρ_{Az} (m)	ρ_{Rg} (m)	Range (km)	Baseline (m)	Δ t (seconds)
		35.10	90.39	205.61	0.1117	0.1016	4.076		
1	07/19/2013	34.99	89.83	206.18	0.1117	0.1016	4.078	7.83	725
		35.02	89.88	249.13	0.1117	0.1016	4.128		
2	07/19/2013	35.03	89.70	249.31	0.1117	0.1016	4.131	2.96	663
		35.05	89.99	249.02	0.1117	0.1016	4.125		
3	07/19/2013	34.98	90.49	248.51	0.1117	0.1016	4.128	5.57	663
		35.05	90.72	181.31	0.1117	0.1016	4.134		
4	07/19/2013	34.88	90.49	181.52	0.1116	0.1016	4.136	12.10	640
		34.97	90.46	205.54	0.1117	0.1016	4.072		
5	07/22/2013	34.97	89.07	206.93	0.1117	0.1016	4.077	4.42	834
		34.98	-89.45	25.46	0.1117	0.1016	4.078		
6	07/22/2013	35.02	-90.29	26.33	0.1117	0.1016	4.085	6.99	838
		35.03	90.38	194.64	0.1117	0.1016	4.134		
7	07/22/2013	35.02	90.17	194.87	0.1117	0.1016	4.137	3.06	678
		35.01	89.80	195.23	0.1117	0.1016	4.134		
8	07/22/2013	35.01	88.91	196.12	0.1117	0.1016	4.135	1.23	677
		34.95	-90.48	69.48	0.1117	0.1016	4.121		
9	07/24/2013	35.12	-91.03	70.05	0.1117	0.1016	4.123	12.53	759
		35.08	91.76	247.24	0.1117	0.1016	4.123		
10	07/24/2013	35.13	90.41	248.60	0.1117	0.1016	4.134	11.96	753
		35.05	-89.88	1.90	0.1117	0.1016	4.128		
11	07/26/2013	35.02	-90.07	2.10	0.1117	0.1016	4.133	5.43	608
		34.94	-89.07	1.09	0.1117	0.1016	4.135		
12	07/26/2013	34.99	-90.78	2.81	0.1117	0.1016	4.130	5.69	608
		34.95	-90.50	26.52	0.1117	0.1016	4.082		
13	07/29/2013	35.01	-90.42	26.44	0.1117	0.1016	4.075	7.73	979
		34.98	-90.30	26.31	0.1117	0.1016	4.080		
14	07/29/2013	34.97	-89.42	25.43	0.1117	0.1016	4.076	4.87	979
		33.52	-89.64	125.63	0.1015	0.1016	4.113		
15	03/27/2015	33.45	88.03	307.94	0.1015	0.1016	4.133	21.19	221
		33.68	-91.60	176.29	0.1015	0.1016	4.147		
16	03/27/2015	33.44	88.45	356.21	0.1015	0.1016	4.125	27.77	242
		33.50	-89.10	224.12	0.1015	0.1016	4.095		
17	03/27/2015	33.48	89.24	45.74	0.1015	0.1016	4.099	3.63	216
		33.48	-90.59	0.62	0.1015	0.1016	3.957		
18	03/27/2015	33.47	90.78	179.24	0.1015	0.1016	3.967	10.42	204