Prepared for submission to *SPIE Defense, Security, and Sensing Symposium,* Anaheim, CA, April 9-13, 2017.
Not yet approved for public release.

SAND2017-1320C

Open Set Recognition of Aircraft in Aerial Imagery using Synthetic Template Models[1]

Aleksander B. Bapst[a,2], Jonathan Tran[a,2], Mark W. Koch[a], Mary M. Moya[a], Robert Swahn[b]
[a]Sandia National Laboratories[3], PO Box 5800, MS 1163, Albuquerque, NM 87185-1163
[b]Defense Threat Reduction Agency, 8725 John J. Kingman Road, Stop 6201, Fort Belvoir, VA 22060-6201

ABSTRACT

Fast, accurate and robust automatic target recognition (ATR) in optical aerial imagery can provide game-changing advantages to military commanders and personnel. ATR algorithms must reject non-targets with a high degree of confidence in a world with an infinite number of possible input images. Furthermore, they must learn to recognize new targets without requiring massive data collections. Whereas most machine learning algorithms classify data in a closed set manner by mapping inputs to a fixed set of training classes, open set recognizers incorporate constraints that allow for inputs to be labelled as unknown. We have adapted two template-based open set recognizers to use computer generated synthetic images of military aircraft as training data, to provide a baseline for military-grade ATR: (1) a frequentist approach based on probabilistic fusion of extracted image features, and (2) an open set extension to the one-class support vector machine (SVM). These algorithms both use histograms of oriented gradients (HOG) as features as well as artificial augmentation of both real and synthetic image chips to take advantage of minimal training data. Our results show that open set recognizers trained with synthetic data and tested with real data can successfully discriminate real target inputs from non-targets. However, there is still a requirement for some knowledge of the real target in order to calibrate the relationship between synthetic template and target score distributions. We conclude by proposing algorithm modifications that may improve the ability of synthetic data to represent real data.

Keywords: automatic target recognition, open set recognition, synthetic data, data augmentation, aerial imagery, histogram of oriented gradients, support vector machine.

1. INTRODUCTION

The role of optical aerial imagery in defense and surveillance is growing in importance, particularly with increasing use of unmanned aerial vehicles for reconnaissance. Rapid and accurate target recognition could aid analysts and warfighters in detecting, identifying and responding to threats. In particular, classification models that make hard assignments from a fixed set of classes and lack a robust ability to reject targets as unknown will not meet these strict requirements. Instead, ATR systems require an extensible open set recognition model, which can add new target classes to an existing model on demand and which incorporate a "don't know" designation for inputs that differ from all known classes.

Military applications often lack comprehensive descriptions of new targets, which precludes data-intensive ATR training. This paper describes the development of open set target recognition for optical images of aircraft using synthetic data to augment or fully replace training data in the absence of sufficient real-world target examples.

[1] 1 This work was supported by the Defense Threat Reduction Agency at Sandia National Laboratories. For additional information, please contact
Mary Moya, Ph.D., mmmoya@sandia.gov.
[2] Authors made equal contributions.
[3] Sandia National Laboratories is a multi-mission laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

1.1 Open set target recognition

Many researchers and commercial sites use deep-learning networks to solve closed set image classification problems[1-4]. However, limitations of deep-learning networks[5] preclude their application to military systems, which will require predictable recognition in the presence of unpredictable inputs. For military systems, open set recognizers[6, 7] incorporate appropriate constraints that generate predictable performance in the presence of unknown inputs. Closed set classifiers cannot rely on training data to represent the full range of all possible inputs. For example, a linear binary SVM trained to discriminate between two classes learns a hyperplane that divides the feature space into two infinitely large regions. The two training classes may only occupy a finite portion of feature space. An input that belongs to neither of the trained classes will produce an incorrect classification with this model. Even training with massive data sets cannot prevent open decision boundaries from producing unpredictable outputs in response to unexpected inputs.

Target recognition systems must respond to three types of types of data classes[8]:

(1) *Known (target) classes*; data that are labeled as positive training examples.

(2) *Known unknown (non-target) classes*; data that are labeled as negative training examples.

(3) *Unknown unknown (non-target) classes*; classes that are not seen during training

To reject all inputs outside of (1), an open set ATR algorithm should incorporate intra-class similarity modeling[9].

Researchers aim to develop open set recognition algorithms that build robust models of known classes and make a declaration or rejection based on a similarity measure between the target and the models, i.e., that learn how to answer the question "*how different is this example from the classes that I know about*?" However, because models are frequently imperfect, including *known unknowns* in the training can facilitate selecting thresholds or refining the model to improve performance.

This paper investigates two open set algorithms using synthetic data for training the known classes, with the goal of precisely defining the requirements on the types of training data required to build a strong classifier. In particular, if using synthetic data imposes an additional constraint; can artificially generated training examples sufficiently represent a known class, and if not, what additional data are needed?

Simonson has developed a statistical approach for open set recognition called *probabilistic fusion* (PF) that combines evidence from multiple sources to produce a single test statistic whose distribution becomes a template model for each class[10]. This frames the problem as a hypothesis test, in which the PF recognizer compares test inputs to the model distributions for each class to obtain a set of p-values and applies thresholds to assign membership to a single class, a subset of classes, or no classes. In the case of multi-dimensional data, each feature provides an individual source of evidence for probabilistic fusion.

Another method of solving the open set problem incorporates a distance-based model based on post-recognition score analysis[7, 11]. Scheirer et al. have described a method of calibrating *meta-recognition* models using an application of extreme value theory for applying a Weibull fit to the tails of output score distributions. The method assesses the accuracy of test inputs by thresholding the cumulative distribution function of the resulting fits, rejecting as unknown examples that are far away from any known class. The Weibull-calibrated SVM (W-SVM) uses meta-recognition models to improve the open set performance of SVMs beyond the state-of-the-art[8]. A similar meta-recognition model based on Weibull tail fitting has also improved the output of convolutional neural networks[6].

1.2 Synthetic Data Generation

Denied target situations and inflexible or expensive data collection methods can preclude recognizer methods that depend on training with large data sets. Instead, we train with images generated from 3D CAD models, which we call synthetic data. Manipulating the CAD model to have any orientation, illumination, background, viewing angle, etc. enables a rich variety of synthetic images.

Research in use of CAD models for computer vision traces its roots back to robotic inspection and object recognition[12-14], which address closed set classification. Khan et. al, provide an example of 3D-model-based closed set classification with HOG features[15]. Greenhalgh and Mirmehdi develop a Gaussian-kernel-SVM with HOG features for model-based open set recognition of road signs[16]. In this paper, we explore the viability of using synthetic data to train two recognizers explicitly designed for open set recognition and test with real target images.

2. TRAINING DATA AND FEATURE EXTRACTION

We collected aerial photographs of various airports and military bases from Google Earth and labeled the locations of aircraft of interest with ground-truth bounding boxes. The images had a resolution of approximately 1 ft./pixel. We extracted sample chips from the bounding boxes. For this work, we chose the Lockheed C-130 Hercules, a common wide-body transport used by many air forces around the world, as our primary target of interest. To test the capabilities of the open set recognition approaches, we selected several aircraft confuser classes: Lockheed P-3 Orion, the Ilyushin IL-38, an "other" confuser set consisting of generic aircraft with various shapes and sizes, and a background set that did not contain any aircraft. The C-130, P-3, and IL-38 are all four-engine turboprop aircraft. To evaluate recognizer robustness, we also trained the recognizers with the P-3 as a secondary target. Figure 1 shows representative examples of these classes. We selected the pixel dimensions of the chips (108 by 150) to approximate those of the C-130 target aircraft (98' long, 133' wingspan) including a small edge buffer of error. Tailoring the chip size to match the C-130 imposes an inherent bias in favor of the target because smaller aircraft like the P-3 do not fully fill the chip frame while larger ones (such as passenger jets) are cropped.

We chose to train the recognizers with extracted image chips because we assume that the recognizer is the final stage in a larger system. We assume that a cueing algorithm, which precedes the recognizer, can locate and extract image chips of possible targets from larger images. The tightly fitted bounding box increases the proportion of aircraft features to background features. In addition, we assume that the cueing algorithm can provide a coarse estimate of target orientation. We expect to train multiple recognizer templates to represent major orientations of the target. We train each target template to be robust to small changes in orientation.

2.1 HOG Features

Histogram of oriented gradients (HOG) is a robust, scale-invariant feature extraction method that is sensitive to edges and shape properties of objects in images[17] and is widely used to create robust descriptors. The advantage of HOG over local keypoint-based descriptors such as SIFT[18], SURF[19], BRIEF[20], etc. lies in its ability to control the number of features while accurately describing shape information. However, standard HOG is sensitive to translations and rotations. We build tolerance to small rotations and shifts into the recognizers by including examples in the training set. Differences in illumination and contrast can also affect gradient distributions because an edge's magnitude is factored into the HOG feature and the feature normalization is not perfect. For this paper, we applied Matlab's HOG function to extract features from the image chips. The function divides each chip into either an 8x8 or 16x16 non-overlapping grid of cells. The function accumulates the gradients in each cell, sorts them into 9 orientation bins and then concatenates the cell histograms together to form a single feature vector of length 8x8x9 = 576 or 16x16x9 = 2304. After normalization, we stacked the feature vectors for each target class to form an array of training data. The advantage of using more cells is the ability to gain a finer representation of shape data; however, if the cells are too small then pixel noise may begin to play a disproportionately large role on the feature representation. In addition, more training data will be required to accurately describe the target.

2.2 Synthetic Data Generation

In this study, we first trained the recognizer with optical image chips derived from the C-130 CAD model. We trained a second recognizer for the P-3 to test repeatability and gain confidence in the results observed. Figures 2(a) and (b) show the CAD models of the C-130 and P-3, respectively. We obtained both models from TurboSquid.com, an online repository of 3D models.

We rendered both models against a black background to facilitate segmentation and to eliminate shadows, which can adversely affect HOG features because of contrast variations. Since the majority of aerial images are taken from directly above the target, we used a single plan-view rendering as the starting point for each synthetic model. We augmented the data set to enable a rich distribution of HOG features from each image.

2.3 Data Augmentation

Manipulating a synthetic template to appear like a real target image requires a variety of distortions. Starting with the CAD model, we applied a random combination of translations, rotations, scaling, resolution reductions, and occlusions and then

added Gaussian noise to the background. Because the distortions of an individual test chip are unknown, we chose to train the recognizer with samples drawn from a random distribution of distortions. We set the variation of the distribution wide enough to avoid overfitting the model, but not so wide to avoid over-representing large distortions. We also introduce Gaussian noise to establish nonzero gradients in the HOG features and to approximate real-world images that contain multiple sources of noise and distortion. We introduced down-sampling variations to adjust the image resolution, which varies in the satellite imagery. We introduced scaling variations to simulate slight variations in the real world image chips. Comparing the synthetic image chips in Figures 2 (c) and (d) with the real image chips in Figure 1 justifies the need for data augmentation. Incorporating too much distortion in the training set could create templates that are too generic. For example, if scaling changes are too large, the ability to distinguish between the P-3, C-130, and IL-38 will diminish because they have similar shapes but different scales.

For future work, we could also consider skewing the image geometry and color channel randomization. However, since we use overhead satellite imagery, skewed geometry is less important. Because HOG finds the max of each color dimension and relies heavily on intensity contrast, color variation is also less important.

We also applied distortions to the real data to create a larger test data set while taking care to avoid corrupting the underlying integrity of the features. To increase the test set size, each sample chip was given 4 rotations, 6 translations in the vertical and horizontal directions, and resized to 3 different scales for a multiplication factor of 432 per chip. After augmentation and HOG feature extraction, we randomly selected a set of 10,000 test chips. Table 2 records the numerical ranges of the distortion and augmentation operations applied to the synthetic and real-world image chips. Table 3 shows the initial number of ground-truth chips for each class, and the number of chips after data augmentation.

Table 1. Maximum values of distortions used in synthetic template training and real-world chip testing

Synthetic Chip Distortion Ranges	
Translation	±12 pixels
Rotation	±5 degrees
Scaling	±10%
Down Sampling	1/N, Nϵ(1…6)
Real-World Chip Distortion Ranges	
Translation	(-5, -3, -1, 1, 3, 5) pixels
Rotation	(-3, -1, 1, 3) degrees
Scaling	(90,100,110)%

Table 2. Summary of Synthetic and Real World Image Chips used for Open Set ATR

Real-World Chips		
Class Name	**# Original Chips**	**# Augmented Chips**
Lockheed C-130	221	10,000
Lockheed P-3	166	10,000
Ilyushin Il-38	15	4320
Other Confusers	737	10,000
Background	10	670
Synthetic Chips		
Lockheed C-130	1	2000
Lockheed P-3	1	2000

2.4 Feature Engineering

We tested a number of feature pre-processing techniques to reduce the dimensionality of the feature vector and empirically select those that are valid for a wide range of image augmentations. These techniques were feature quantization, principle component analysis, and feature masking.

2.4.1 Feature Quantization

We applied a quantization scheme to the features that was directly inspired by the template generation procedure for multinomial pattern matching[9]. We define an $M \times N$ array of HOG features where each column contains M training samples and each row contains N features. First we binned the feature columns according to intensity and then normalized the rows by the sum of all bin counts to obtain a feature vector for each profile that sums to 1. This improves the illumination invariance of the HOG features, as different amounts of image blur can result in gradients of varying intensity, but similar orientation. Therefore, assigning bin identities to ranges of intensities may reduce the impact of local fluctuations in gradients. Note that the features are still sensitive to rotation and translation because as an aircraft changes position within an image chip, the activated HOG features will change with the positions of the cells that contain the target object. In this paper, we used 10 equally-space bins for feature quantization.

2.4.2 Principle Component Analysis

We apply PCA dimensionality reduction to extract salient features. The data reduction provided by PCA truncation increases speed of the recognizer training. We test the PCA reduction on open set algorithms by first computing the $N \times d$ eigenvector matrix $V_{template}$ from a set of synthetic HOG template data, where N is the number of features and d is the number of principle components. The template data is first centered by subtracting the mean feature vector $\mu_{template}$. Then, for any $1 \times N$ vector of test features, the data is transformed via the following expression:

$$F_{trans} = \left(F_{class} - \mu_{template}\right) \cdot V_{template} \quad (1)$$

2.4.3 Feature Masking

The motivation behind feature masking is to reduce the number of background features, which don't contain information relevant to the object of interest. We created an object mask for the Lockheed C-130 and P-3 synthetic templates by segmenting the black background from each chip and combining all masks into a master mask that captures all rotations, translations and scales in the augmented data. Figure 3(b) shows that this process produces a mask that is larger than any individual target and should accommodate the majority of relevant features. Next, to find the HOG features that correspond to the mask, we divide the master mask into HOG cells. In each cell, we count the number of pixels that fall on the mask, and, if the count exceeds the 50% threshold, we keep the block of HOG features corresponding to that cell and reject those with smaller percentages. We use the indices corresponding to the mask features to extract valid pixels for all sets of HOG data. Figure 3 shows an example of the mask developed for the synthetic C-130 template.

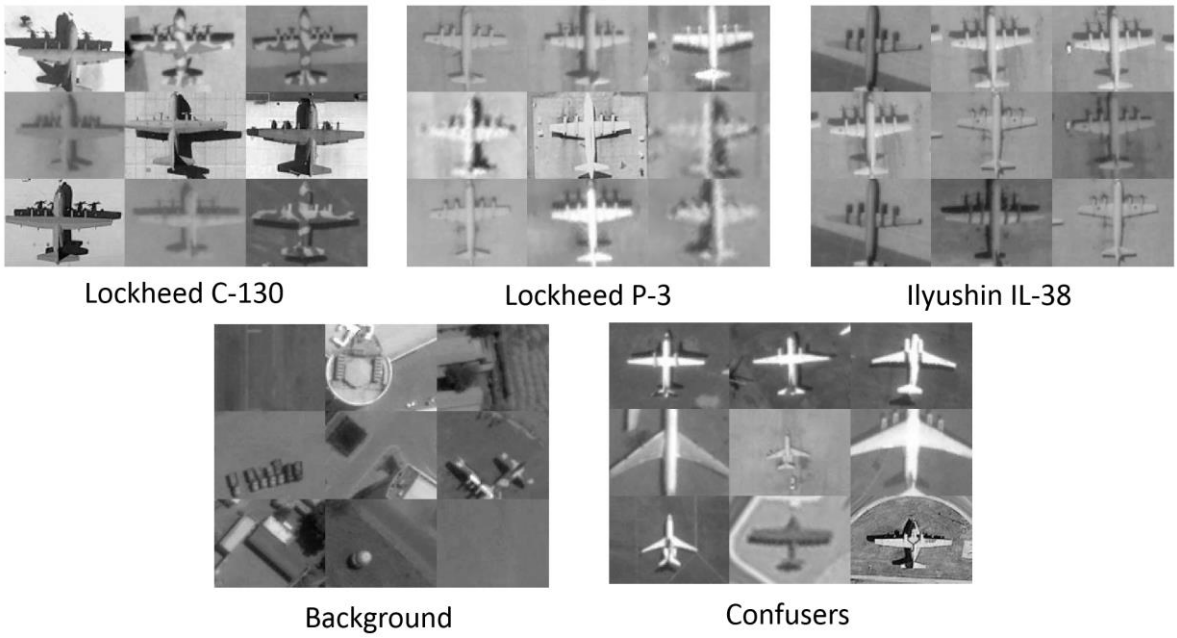

Figure 1. Examples of various aircraft class chips. Note that the Lockheed P-3 and Ilyushin IL-38 are very similar in appearance. Numbers of each chip used: C-130 (221), P-3 (166), IL-38 (15), Background (10), Confusers (737).

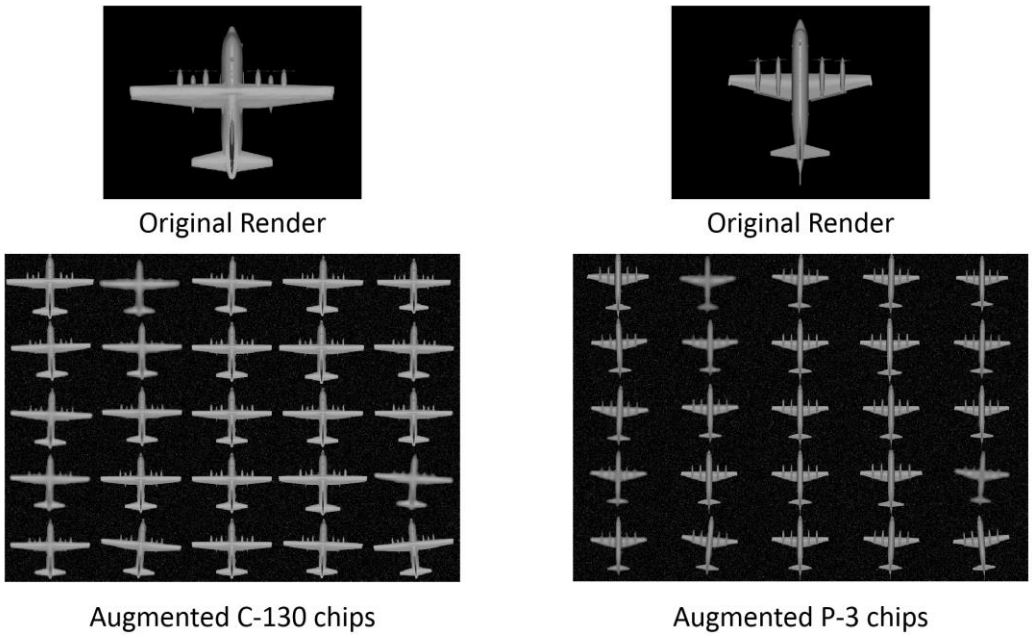

Figure 2. Examples of synthetic data rendered from 3D CAD models and some of the image augmentations that were performed to generate a wide range of simulated HOG features, such as rotation, translation, down sampling, and addition of Gaussian noise.

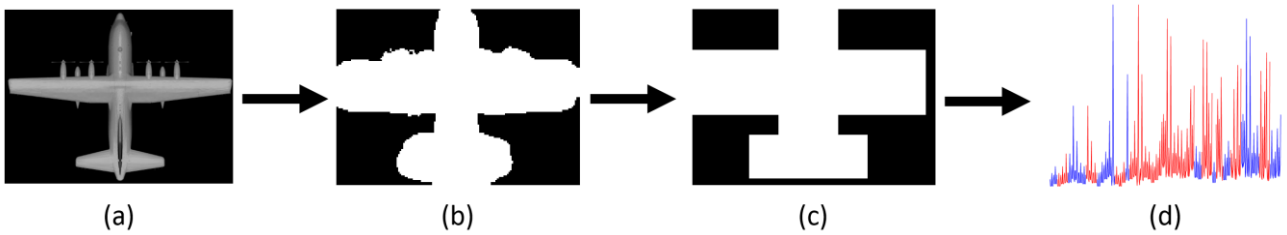

Figure 3. Illustration of feature masking procedure. (a) Original 3D CAD rendering, (b) combined mask from all augmented chips, (c) mask with HOG cells blocked out, (d) HOG features (red) corresponding to unmasked cells

3. PROBABILISTIC FUSION

Probabilistic fusion (PF)[10] is an open set classification technique that forms a combined test statistic from multiple sources of evidence. Each source, such as each feature in a HOG vector, produces a score, represented by the random variable Z_i, $i \in$ [1..N]. The PF algorithm estimates a parametric cumulative distribution function (CDF) $F_i(z_i)$ from target training data such that:

$$F_i(z_i) = Prob(Z_i \leq z_i) \tag{2}$$

We train with target data only. The inherent properties of the CDF guarantee that $F_i(z_i)$ are uniformly distributed over the range [0,1] for matched targets. Non-target features that do not match the target distribution will produce strong concentration of values close to 1. We apply the following mapping to $F_i(z_i)$:

$$Y_i = -\log(1 - F_i(z_i)) \tag{3}$$

Because $F_i(z_i)$ is uniform for targets, Y_i will have a standard exponential distribution for targets and be large for non-targets. To form the fused test statistic, we sum all N mapped scores:

$$S_f = \sum_{i=1}^{N} Y_i \tag{4}$$

If the transformed Y_i are uncorrelated, then the distribution of fusion scores S_f are gamma distributed with shape and scale parameters $r = N$ and $\lambda = 1$, respectively. In the case of features Y_i and Y_j with a nonzero correlation coefficient ρ_{ij}, the gamma parameters can be estimated using the following approximation according to Simonson[10]:

$$C = \sum_{i=1}^{N} \sum_{j \neq i} \rho_{ij} \tag{5}$$

$$\hat{r} = \frac{N^2}{N + C} \tag{6a}$$

$$\hat{\lambda} = \frac{N}{N + C} \tag{6b}$$

3.1 Training procedure

We train a probabilistic fusion model on an $M \times N$ array of HOG features, where each column contains M training samples and each row contains N feature scores. Down each column, we compute the mean feature score, subtract the mean from all samples and then apply the absolute value. This forces the target to have small scores relative to non-targets. We then parametrically fit each feature column with a set of generic parametric distributions: exponential, normal, beta, and gamma.

Prior experience with applying PF to other feature-based recognizers guided our selection of these distributions. For each feature, we apply the Kolmogorov-Smirnov (KS) test for goodness-of-fit, choose the best-fitting distribution and store its parameters in the target model. We use the model distribution for each feature and Equations (2) and (3) to compute $F_i(z_i)$ and Y_i for each feature and each training sample. Equation (4) then sums the transformed features together to produce a single fused score for each training sample. Finally, we use Equations (5) – (6b) to estimate parameters of the gamma distribution that models the fused scores.

3.2 Testing procedure

Given a new test example and a template model trained on a representative set of training data (such as synthetic HOG features), the algorithm maps each feature in the new example to its respective cumulative distribution functions, applies the mapping to produce feature scores and sums the scores to produce the fused score using the procedure in Section 4. It computes the p-value of the fused score against the estimated gamma distribution of the model, and applies a threshold to accept or reject the example as a member of the model class. If multiple models are tested and none are accepted, then the test example is classified as unknown. Otherwise, if multiple models are accepted, one may choose to assign the class label with the largest p-value, or else assign multiple class labels to the example.

4. SVM

SVMs are a popular baseline approach for classification because they can produce good results without needing extensive fine-tuning. In a closed set environment where all the targets and non-targets are known a priori, binary SVMs perform well. However, because they lack a built-in mechanism for rejection of targets, binary SVMs are not applicable to the open set domain. One-class SVMs provide a method for training using only positively labelled targets and can reject targets that lie outside the class decision boundary, but generally they do not perform as well as binary SVMs[7]. The Weibull-calibrated SVM (W-SVM) leverages the advantages of both the binary SVM and the one-class SVM to produce a robust classifier that can reject non-targets that are not known at training time[8]. In in this paper, we compare one-class SVMs, binary SVMs, and W-SVMs to each other and to PF.

4.1 One-class SVM

We chose a one-class SVM for this application because of its ability to recognize targets and reject non-targets without the need for negatively labelled training data. Principally, one-class SVMs, in particular the Schölkopf et. al implementation, attempt to define a hyperplane or hypersurface that maximizes the distance separating the training data from the origin[21]. After mapping into an SVM kernel space, the data point's distance from the boundary of the hyperplane defines the classification score. A highly positive score indicates a data point that lies near the center of the N-dimensional (defined by the number of features) cluster of training data. A highly negative score would indicate a point that is far away from the decision boundary. However, in the SVM kernel space, the nature of radial distance causes ambiguity, illustrated in Figure 4. In this figure, the concentric circles represent points equidistant from the decision boundary, in other words, contours along which SVM scores are equal. Non-targets will have the same scores regardless of the direction in which they diverge. This is usually not problematic as it is not important how a data point diverges from the boundary, but in this particular application, we find that certain features may dominate the decision causing seemingly dissimilar targets to look similar in the SVM kernel space. An example of this will be illustrated in Section 5.2. As a result of this ambiguity, we chose to explore other methods of discrimination such as the Weibull-calibrated SVM that should have different decision boundary regions.

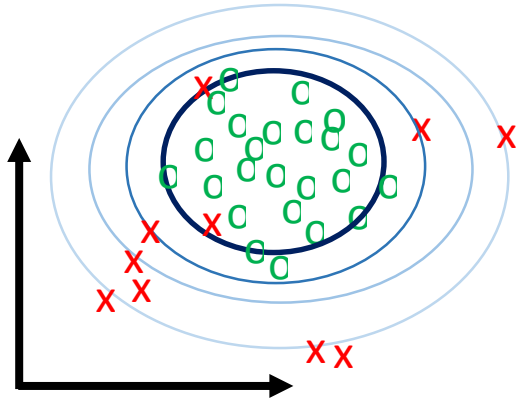

Figure 4. Two-dimensional representation illustrating ambiguity in one-class SVM with green O's indicating in class targets and red X's indicating out of class non targets

In this study, we trained an SVM in Matlab using 2000 synthetic target samples of HOG features. Our one-class SVM uses the radial basis function (RBF) kernel as defined by the following equation:

$$k(\boldsymbol{x}, \boldsymbol{x}') = e^{-\frac{\left||\boldsymbol{x}-\boldsymbol{x}'|\right|^2}{2\nu^2}} \tag{7}$$

which includes one parameter, ν. This parameter controls the amount of training data that can be considered outliers. As this parameter shrinks, the in-class region grows, capturing more of the training data, but also generating more false alarms. When there are differences between the training data and the test data, such as the case when we train with simulated data, we need a smaller value of ν. For this paper, the ν parameter was set to 0.01.

4.2 Binary SVM

A binary SVM trains on both target and non-target data. In order to augment the non-target training set to exploit the discrimination power of the binary SVM (since only two aircraft CAD models were used here), we added a subset of the various confuser aircraft to the non-target set. We reserved the remainder of the confuser set and the other identified aircraft for testing. To extend and bridge the advantages of the two SVM types, we implemented and tested the W-SVM below.

4.3 W-SVM

The W-SVM uses the Extreme Value Theorem (EVT) to test the hypothesis that a Weibull distribution models the tails of the class score distributions[8]. In theory, the distributions model the probability, P_η, that a sample is a member of the target set and the probability, P_ψ, that a sample is not a non-target, by sampling the tails of the target and non-target scores, respectively. These two sources of evidence are then combined to form the W-SVM recognition score for an unknown input.

Fundamentally, the W-SVM is a binary SVM with a one-class SVM precursor to reject obvious sample chips that don't resemble the target and, thus, limits the problem of the infinite classification spaces of the binary SVM. We applied the following W-SVM algorithm:

1. Train a one-class SVM on the synthetic data.
2. Fit a Weibull CDF to the smallest (most strongly classified) scores of the target data.
    a. Apply a maximum-likelihood estimation (MLE) to fit the Weibull parameters to the one-class SVM scores.
    b. Set a small threshold, σ_T, on the one-class SVM Weibull scores to capture a wide swath of targets; subsequent binary SVM should remove excess non-targets.
    c. Define a binary indicator, ι_o that is set to 1 if the one-class SVM score is greater than σ_T and 0 otherwise.
3. Train a binary SVM with real training data and real non-target data.
4. Fit a pair of Weibull CDFs to the smallest target scores and the largest non-target scores.
    a. Most parameter estimators require non-negative scores so we used an offset to make all scores positive.

b. The largest non-target scores are the values closest to the decision boundary.

5. Using the two CDFs, obtain a probability score from each distribution (based on the samples binary SVM score), compute a new probability and keep or reject the sample based on the binary indicator obtained in step 2.

W-SVM can then be implemented using the Equations (8) and (9) (listed below).

$$P_{match}(x) < \sigma_T, \iota_{match} = 0;\ \iota_{match} = 1\ otherwise \tag{8}$$

$$P_{WSVM} = P_\eta \times\ P_\psi \times \iota_{match} \tag{9}$$

Equation (8) gives the conditioning for W-SVM as mentioned in step 2 of the algorithm described above. For a given sample, x, a probability that it matches the one class training, P_{match}, is obtained from the Weibull CDF fit of the one-class tail data. The threshold, σ_T, to limit the allowable samples is then defined according to the data.

Equation (9) defines the score used for W-SVM. P_η is the probability obtained from the target Weibull CDF for the sample, x, given its binary SVM score. P_ψ is the probability obtained from the non-target Weibull CDF for the same sample, x. Combining the result from Equation (8) then imposes the closed boundary learned from target-only training, which limits the open nature of the binary SVM and prevents it from extending to infinity away from the boundaries.

In our application, there is significant overlap in the SVM scores of the non-match samples and the targets as well as a large separation between the synthetic training data and real-world sample scores. Details of this can be seen in Section 5.2. The effect of this is real data is necessary to find the threshold, σ_T and the Weibull fits for the binary SVM tails. Because the synthetic data scores are separated from the real-world target scores, some of the target data produces negative scores. To find a threshold that limits the amount of samples that could confuse the binary SVM but keep the majority of the targets, a number of real world targets is necessary. In addition, the binary SVM trained on synthetic data and real-world non-match data results in target scores that are not only negative but highly overlapped with other non-match data. If Weibulls are fit on the synthetic and non-match tail scores, all of the target data will have very low probabilities offering no new information. However, if the Weibull is fit to the target data as well as the non-target data, we can obtain probabilities for new class samples that fall between these two distributions in the binary SVM space.

5. EXPERIMENTAL RESULTS AND DISCUSSION

This sections shows the results of training with synthetic data and testing with real data. As a point of comparison we also show the results the of training and testing with real data. This is for comparison purposes only, as the objective of this paper is to train with a denied target described by synthetic data. In these experiments, two open-set aircraft classifiers were developed one for the C-130 and other for the P-3. For each experiment, one target class was used as the training template and the other was used as an additional confuser class along with IL-38, and background chips.

The combination of data pre-processing parameters that resulted in the best separation between classes for both probabilistic fusion and SVMs was a HOG with a block size of 8x8 cells (576 features total) followed by PCA on the template data and projection of all other features onto the top 100 principle components. We show results using two visualizations. The first method plots the resulting score distributions for each classifier using nonparametric density estimation with a Gaussian kernel, and the second method plots the receiver operating characteristic (ROC) curves between the real world target distribution and the other real world classes (confusers, background, etc.).

5.1 Probabilistic Fusion

Using the synthetic data, we generated probabilistic fusion templates for the C-130 and P-3 and then tested with augmented real-world data. To test the null hypothesis that there shouldn't be any difference between training with synthetic data and training with real data, we also generated templates for real-world C-130 and P-3 data using a randomized 20% split on the training data, with the remaining 80% used for testing.

The fused score distributions for real-world templates (Figure 5, (a) and (b)) show that the corresponding real-world data distribution aligns well with the distribution of the template, while remaining separated from other aircraft classes. If we

approximate the template distribution by a gamma distribution using Equations 5 - 6(b), we can build an open set recognizer that only requires known target data to build the template. Because the non-target distributions are separated from the target, the recognizer can reject non-targets based on a p-value criterion. Figure 6, (a) and (b) show the empirically generated ROC curves that demonstrate its open set performance.

For the synthetic models, the template distributions do not align with their corresponding real-world distributions, and are located far to the left of the main cluster of real-world class data (Figure 5, (c) and (d)). The synthetic template model can still separate real-world targets from various other aircraft classes and background, and Figure 6, (c) and (d) shows that a threshold-based recognizer can be created. However, additional knowledge of the real-world template is required to define the relationship between the synthetic and real-world data.

In all cases, the real world target score distribution was closest to its respective template, while the confuser and background distributions were located furthest away. In addition, the P-3 and Il-38 distributions were very close to each other. This indicates that if an open set classifier were constructed, it would not be able to distinguish between these two aircraft. This makes intuitive sense and may be acceptable in practice, as the two aircraft are very similar in appearance, even to the human eye. The results suggest that probabilistic fusion is viable as an open set target recognition algorithm, provided that real-world data is used to build the individual class templates. This indicates that further research is required to create synthetic data that resembles real data or to determine the transformation between the synthetic scores and the real-world scores. Some suggestions in this direction are included in the Summary and Future Work section.

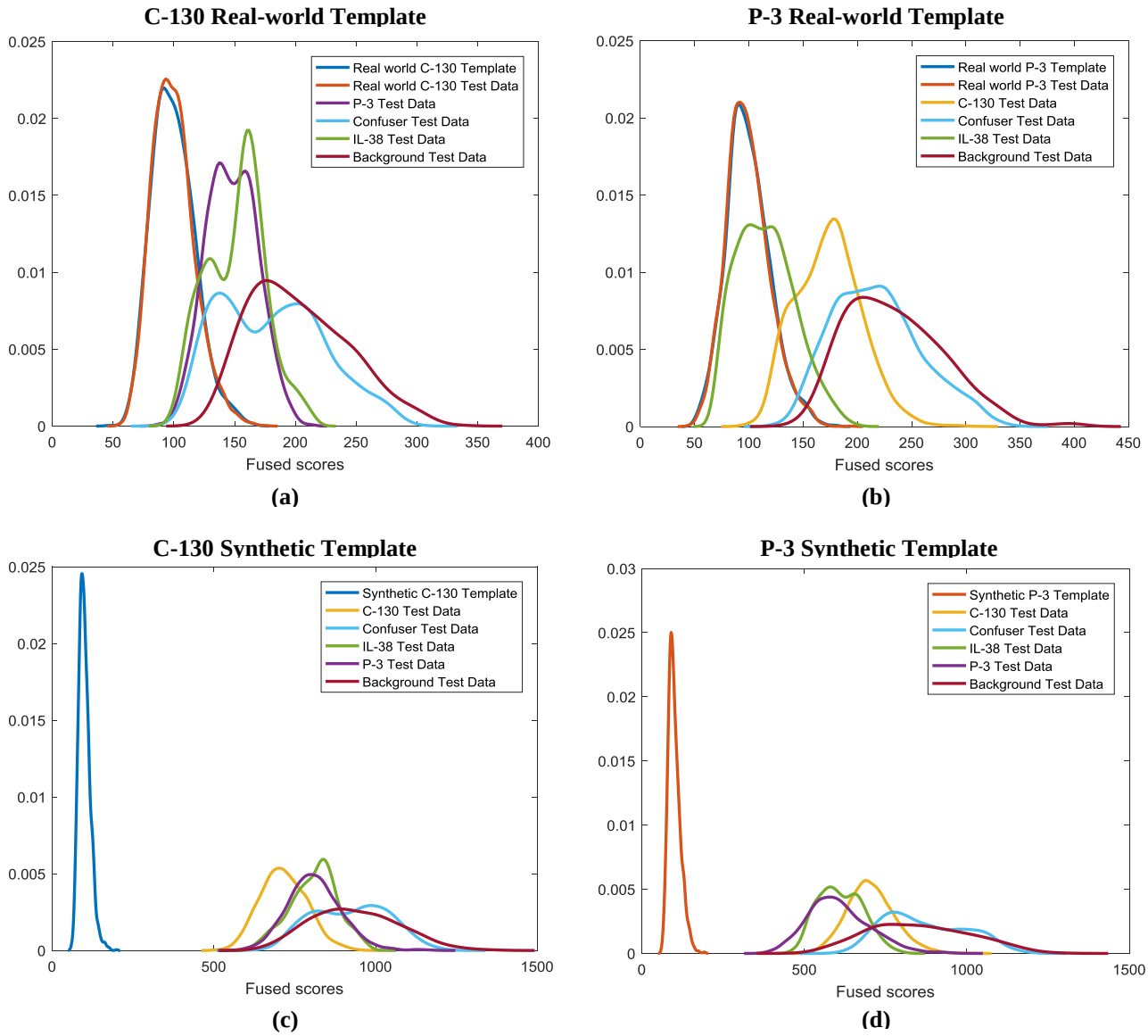

Figure 5. Distributions of fused scores for aircraft HOG data against C-130 and P-3 synthetic and real templates. In each case, the target of interest is the class that corresponds to the template distribution. (a) Trained on real C-130 data. (b) Trained on real P-3 data. (c) and (d) trained synthetic data for the C-130 and P-3, respectively.

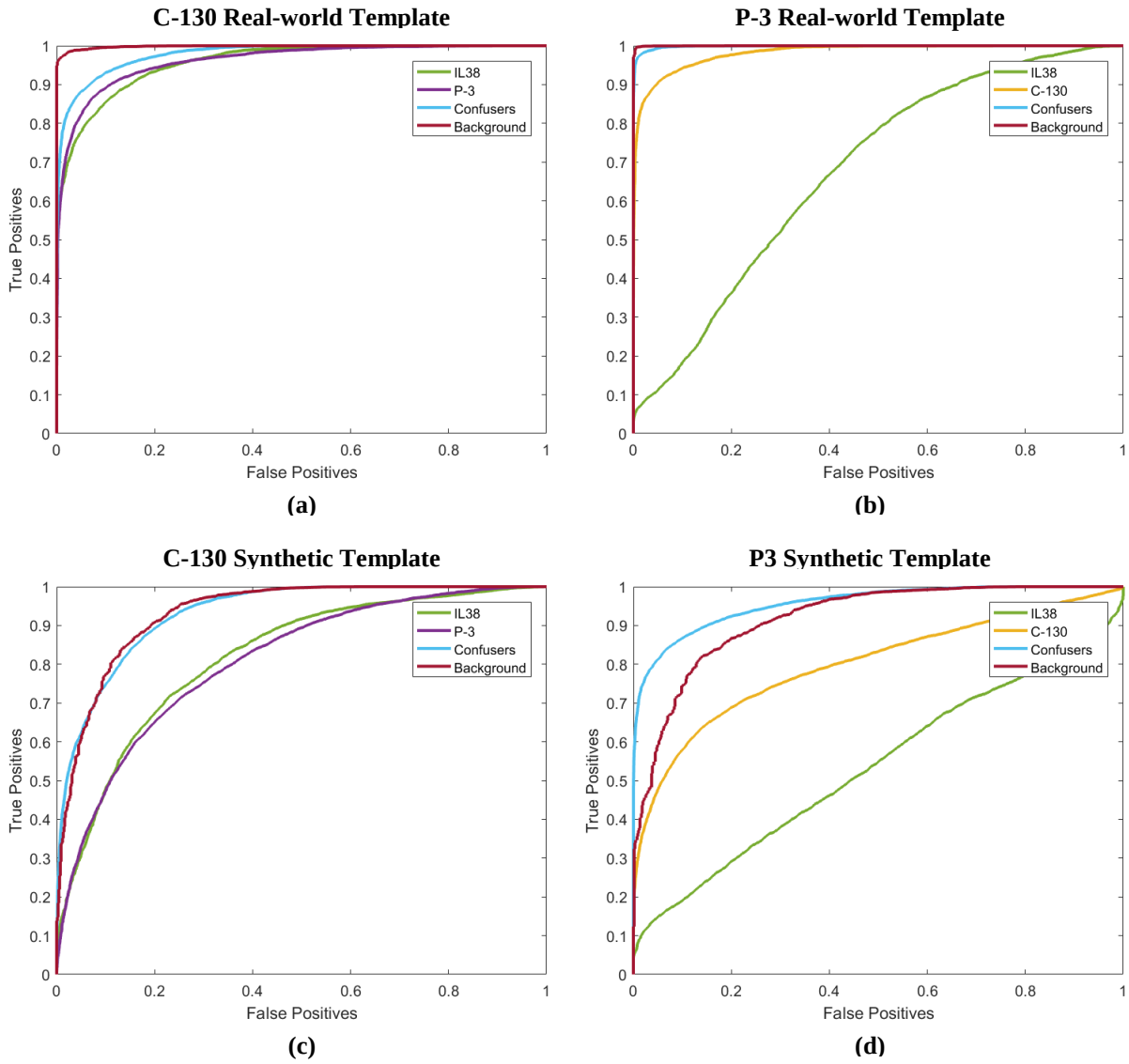

Figure 6. ROC curves of fused scores for real-world C-130 and P-3 score distributions against other aircraft classes. The title of each plot refers to the template used to generate the distributions. (a) Trained on real C-130 data. (b) Trained on real P-3 data. (c) and (d) trained synthetic data for the C-130 and P-3, respectively.

5.2 SVM

We prepared the data for SVM evaluation in a similar fashion as probabilistic fusion. However, because the SVM requires more training data than probabilistic fusion, we used 80% of the real data for training and 20% for testing, as opposed to the opposite for PF.

Figure 7(c) shows that the one-class SVM trained with synthetic data produces separation between the C-130 target test chips and the non-target test chips, including P-3, IL38, other planes and background. In the one-class SVM score space, there is enough separation between the target and non-target samples such that an operating point can be found that gives a reasonable probability of detection and false alarm rate. However, when trained with synthetic P-3 data, the one-class SVM fails to separate the target P-3 test chips from the non-target C-130 test chips, as seen in Figure 7(d), an example where the one-class SVMs could produce ambiguity and false positives on an unknown class. Even if real samples were available to find an operating point, the classifier would falsely identify C-130s at the same rate as the target in this space.

Even though the SVM trained on synthetic data can discriminate test targets from non-targets, the real score distributions do not align well with the synthetic score distributions. As with PF, for scenarios where real data is not available, estimating a reasonable operating point may be difficult.

W-SVM incorporates a binary SVM in conjunction with the one-class SVM. We trained a binary SVM to separate synthetic target data from *known unknowns*, or the set of confuser aircraft. The binary SVM better separates targets from confuser aircraft than does one-class SVM. However, the binary SVM produced more overlap between the score distributions for the four-prop military aircraft. Following the normal W-SVM procedure, the smallest scores from the training distributions are used to find a Weibull fit. However, since the scores overlap to such a high degree, no meaningful declaration can be made, except to reject the outlying confuser aircraft. To get a usable solution, we need to estimate the distribution tail of the match scores from real data. This provides a proper offset and threshold as mentioned in the W-SVM algorithm steps 2b. and 4a. outlined in Section 4.3. While generic confusers should generally be available for training, target sample chips may not. For the sake of comparison, we assume that if real target data are available, W-SVM provides a solution in the open set domain that could be explored.

In order to assess the potential of the algorithms as open set recognizers, we generated ROC curves for the one-class SVM, binary SVM, and W-SVM algorithms trained on synthetic template data (Figure 8). All of the different approaches generally perform very well when discriminating "other" confuser targets. However, with visually similar military aircraft or uncategorized background chips, the performance is more varied. Overall, probabilistic fusion outperforms the SVMs with the exception of IL-38 in a couple of scenarios.

The W-SVM performed only marginally better than a binary SVM using synthetic data, as seen in Figure 8 (c) – (f). As suggested by Scheirer[8], the W-SVM approach does appear to limit the out-of-class data, as evidenced by the upper limits on true positive rate seen in the ROC curves, 8 (e) – (f). However, with sample chips that were visually similar but still unique, it only slightly outperformed the binary SVM and in some cases underperformed against the one-class SVM. This behavior likely results from the similarity of the different data types in the HOG feature space and the separation of the synthetic training template and the real-world targets. Expanding the one-class threshold, σ_T would capture a larger number of test targets and increase the probability of detection (PD), but, because of the overlapping target/non-target distributions, also would capture more unknown unknowns. Widening the target-capture space in this way causes the W-SVM to devolve to a binary SVM with the one class SVM offering no additional discrimination.

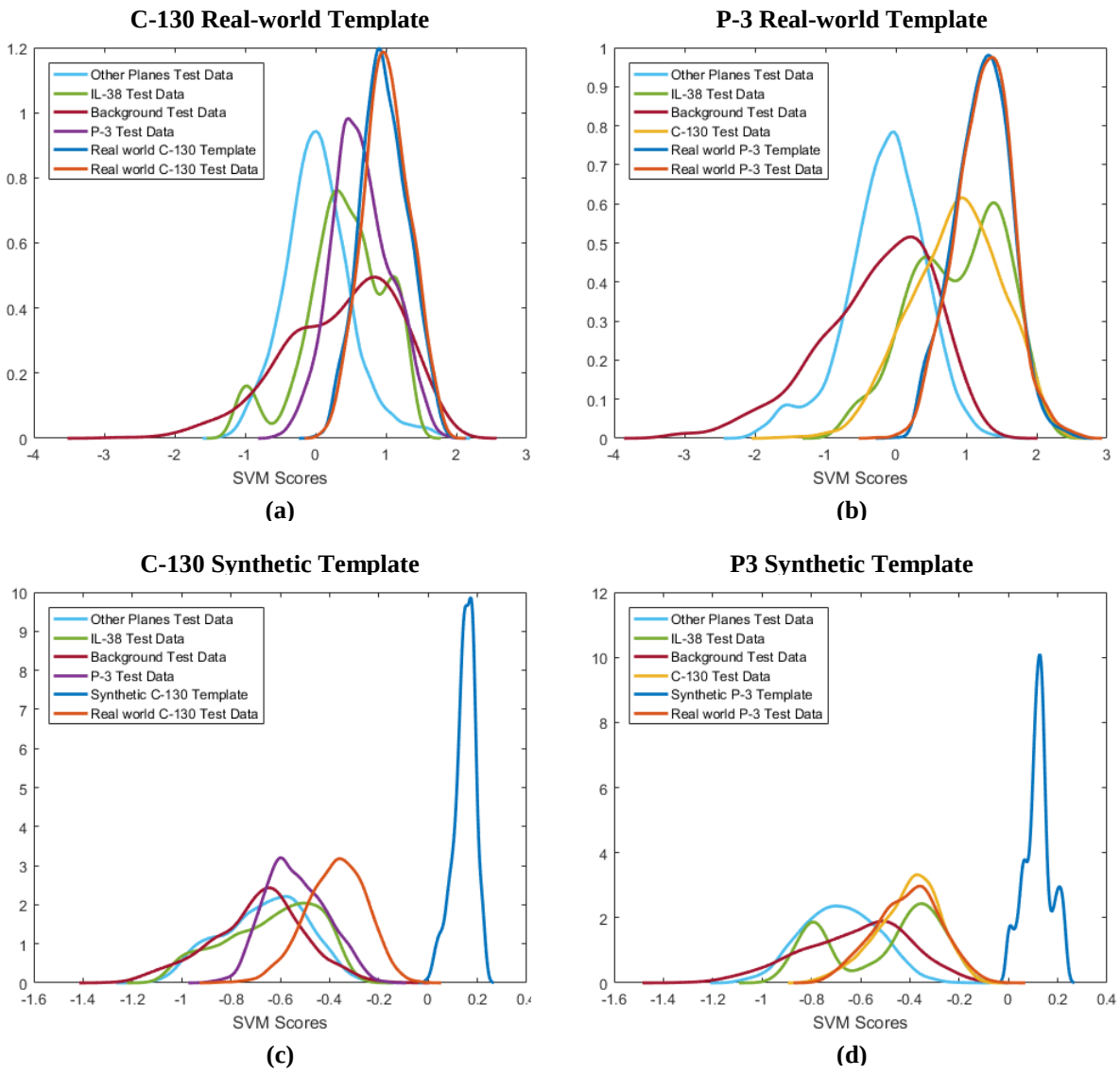

Figure 7. One-class SVM score distributions using various training templates (a) Training using real-world C-130 samples (b) Training using real-world P-3 samples (c) Training using synthetic C-130 samples (d) Training using synthetic P-3 samples

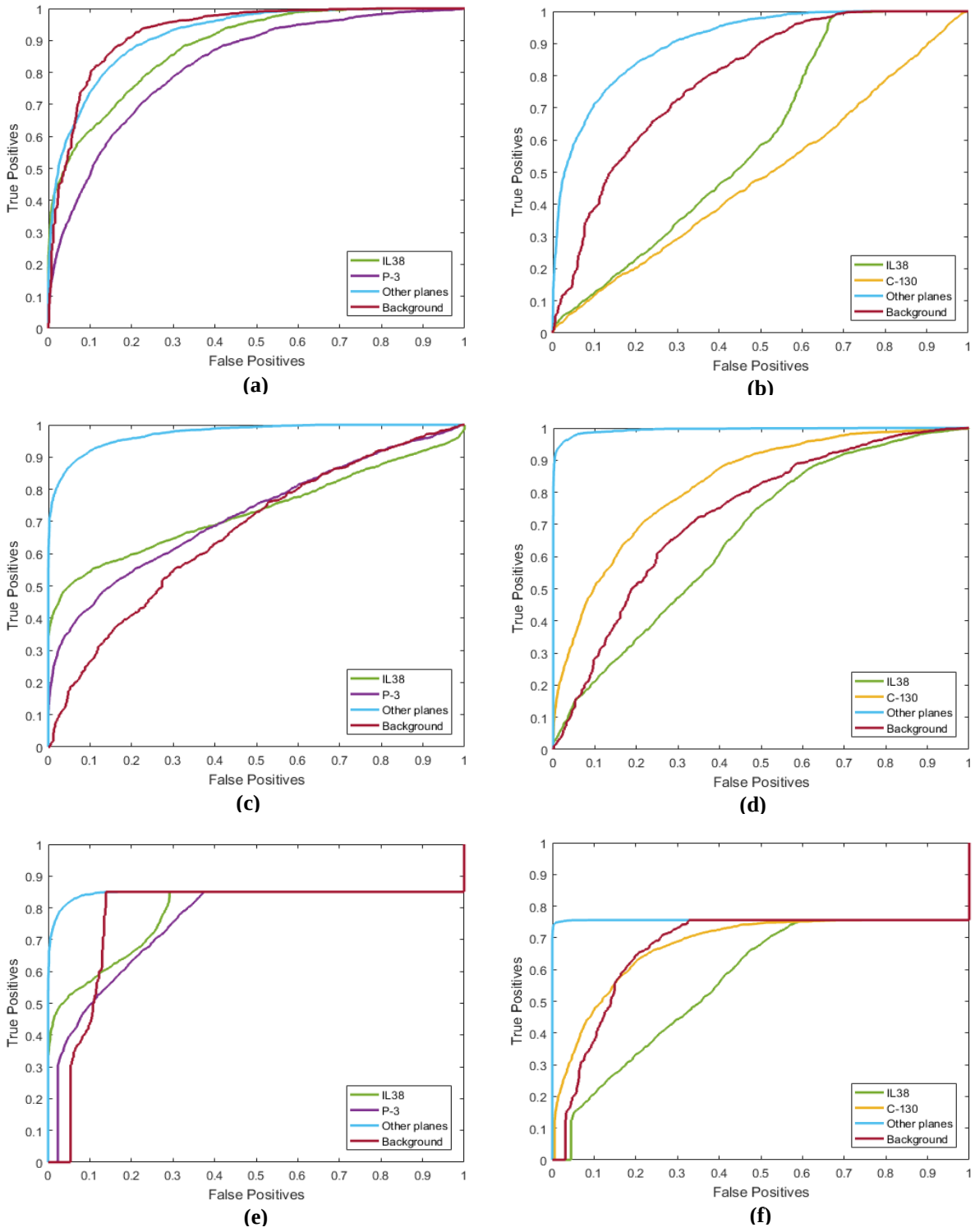

Figure 8. ROC curves for (a) one-class SVM with synthetic C-130 template, (b) one-class SVM with synthetic P-3 template, (c) binary SVM with synthetic C-130 template, (d) binary SVM with synthetic P-3 template, (e) W-SVM with synthetic C-130 template, (f) W-SVM with synthetic P-3 template.

6. SUMMARY AND FUTURE WORK

Using synthetic data to train and classify real-world data has many advantages including reduced cost, ease in acquiring a variety of imaging geometries, ability to replicate a large number of scenarios, and a potentially unlimited number of training samples to use. In this paper, we have shown that the recognizer score distributions generated by synthetic templates are separable and satisfy the requirements for an open set algorithm, as long as some knowledge of the real template is available to calibrate the separation between synthetic and real distributions of scores. Using synthetic data alone to build a robust open set recognizer remains an open problem. Furthermore, even if only a small amount of data is available, data augmentation can be used to create larger and more diverse data sets, reducing the reliance of the algorithm on real world data.

In the future, we propose several avenues of investigation that could reduce or eliminate the need for real-world samples. A potential area of interest may be to learn a transformation between the real target scores and synthetic scores in an unsupervised fashion, in order to calibrate recognition algorithms without the need for knowledge of the real target.

In this paper, HOG features were used because of their ability to generate a compact, well-defined representation of an entire scene. Another global scene descriptor that may be useful is the GIST method, which models the structure of a scene by convolution with Gabor filters at various scales and orientations[22]. GIST descriptors offer an alternate method of describing a scene, but like HOG, are still not invariant to rotation and translation.

Another possibility is to use alternate feature extraction methods that may better tolerate rotational and translational variance, such as using scattering operators for image recognition[23]. We also believe that our method to augment synthetic data may not be robust enough to represent the variations present in real data. In particular, the presence in real images of large shadows dependent on the sun angle may add an additional source of confusion for recognition algorithms due to the sensitivity of HOG features to large contrast gradients. Developing a technique to suppress shadows may improve recognizer performance. By creating a more representative data set using different backgrounds, distortions, illuminations, or imaging geometries it will be possible to generate data that generalizes better to the real world. One such approach may be to use a generative convolutional neural network approach such as Google's Deep Dream to augment synthetic data by enhancing it with patterns learned from real images[24]. We believe that further research in this area will narrow the gap between synthetic and real world training data and open up new avenues in machine learning in which neither representative "known unknowns" nor expensive data collections are required.

ACKNOWLEDGEMENTS

The authors wish to thank Samuel A. Bolin for extracting the aerial images used in this paper and Darren Rodriguez for providing ground truth labelling of aircraft. We are grateful to David Yocky for insights, encouragement and support.

REFERENCES

[1] G. Hinton, O. Simon, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural computation, 18(7), 1527-1554 (2006).
[2] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision." 253-256.
[3] D.-A. Clevert, and T. H. Unterthiner, Sepp, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," arXiv:1511.07289, (2015).
[4] B. Graham, "Spatially-sparse convolutional neural networks," arXiv:1409.6070, (2014).
[5] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images." 427-436.
[6] A. Bendale, and T. Boult, "Towards Open Set Deep Networks," Computer Vision and Pattern Recognition, (2015).
[7] W. Scheirer, A. Rocha, A. Sapkota *et al.*, "Towards Open Set Recognition," IEEE Trans. Pattern Anal. Mach. Intell., 35(7), 1757-1772 (2013).
[8] W. J. Scheirer, L. P. Jain, and T. E. Boult, "Probability Models for Open Set Recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(11), 2317-2324 (2014).
[9] M. L. Koudelka, J. A. Richards, and M. W. Koch, "Multinomial pattern matching for high range resolution radar profiles," Algorithms for Synthetic Aperture Radar Imagery XIV, 6568, (2007).
[10] K. M. Simonson, [Probabilistic Fusion of ATR Results] Sandia National Laboratories, (1998).
[11] W. J. Scheirer, A. Rocha, R. J. Micheals *et al.*, "Meta-Recognition: The Theory and Practice of Recognition Score Analysis," IEEE Trans. Pattern Anal. Mach. Intell., 33(8), 1689-1695 (2011).
[12] R. Hoffman, H. R. Keshavan, and F. Towfiq, "CAD-driven machine vision," IEEE Trans. Systems, Man, and Cybernetics, 19, 1477-1488 (1989).
[13] M. Trivedi, C. Chen, and S. B. Marapane, "A Vision System for Robotic Inspection and Manipulation," Computer, 22(6), 91-97 (1989).
[14] B. Bhanu, "CAD-based robot vision," Computer, 20(8), 12-16 (1987).
[15] S. M. Khan, H. Cheng, D. Matthies *et al.*, "3D model based vehicle classification in aerial imagery." 1681-1687.
[16] J. Greenhalgh, and M. Mirmehdi, "Real-Time Detection and Recognition of Road Traffic Signs," IEEE Transactions on Intelligent Transportation Systems, 13(4), 1498-1506 (2012).
[17] N. Dalal, and B. Triggs, [Histograms of Oriented Gradients for Human Detection] IEEE Computer Society, (2005).
[18] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, 60(2), 91-110 (2004).
[19] H. Bay, A. Ess, T. Tuytelaars *et al.*, "Speeded-Up Robust Features (SURF)," Comput. Vis. Image Underst., 110(3), 346-359 (2008).
[20] M. Calonder, V. Lepetit, C. Strecha *et al.*, [BRIEF: binary robust independent elementary features] Springer-Verlag, Heraklion, Crete, Greece(2010).
[21] B. Schölkopf, R. C. Williamson, A. J. Smola *et al.*, "Support Vector Method for Novelty Detection." 12, 582-588.
[22] A. Oliva, and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," International Journal of Computer Vision, 42(3), 145-175 (2001).
[23] J. B. Estrach, [Scattering Representations for Recognition] Ecole Polytechnique, (2012).
[24] A. Mordvintsev, C. Olah, and M. Tyka, [Inceptionism: Going Deeper into Neural Networks] Google Research Blog, (June 17, 2015).