Identifying Anomalous Objects in SAS Imagery Using Uncertainty

Calum Blair* John Thompson*
* Institute for Digital Communications
University of Edinburgh
Edinburgh, UK
Email: c.blair@ed.ac.uk

Neil M. Robertson†
†Institute for Sensors, Signals and Systems
Heriot-Watt University
Edinburgh, UK

***Abstract*—Object detection in modalities such as synthetic aperture sonar (SAS) is affected by the difficulty of acquiring a large number of training samples. If object classes not present in the training dataset are detected during testing, they can be mis-classified as one of the training classes. This increases overall false alarm rate and affects operator reliability and trust in the detection algorithm. Previous work showed that classification algorithms are often overconfident in their predictions and hence cannot reliably flag image regions about which the algorithm is uncertain or which need further sampling or processing. This paper describes object detectors based on SVMs and Gaussian Processes for SAS imagery, followed by probabilistic calibration of detector confidence scores. The entropy or uncertainty of these scores is then used to identify low-confidence regions and indicate the presence of previously unseen or anomalous objects.**

I. Introduction

Many algorithms for detection of objects in challenging environments rely on the use of training data involving a large set of positive and negative samples. These algorithms are used to perform automatic classification and detection. In situations where either (i) a limited number of samples are available for training or (ii) objects present in the test set are from classes not represented in the training set, misclassifications of test samples may occur. In this paper we study synthetic aperture sonar (SAS) imagery [1]. Collection of images like this is expensive in terms of equipment and time; this limits the quantity of data which can be used for training, and hence the robustness of individual detectors.

SAS is often deployed from unmanned platforms. Ideally these would be as autonomous as possible, capable of making decisions based on the environment that they perceive and detections within it. In these cases, decision making is hampered by classifiers which are overconfident in their predictions [2]. In classifier training, the usual objective is to minimise misclassifications (false positives and negatives), and reliable indications of uncertain classifications are less of a priority. We use existing definitions of *accuracy* (the proportion of samples which are classified correctly) and *reliability* (how well the classifier's confidence prediction agrees with ground truth observations) [3]. These two quantities are not the same and for a given problem, the most accurate detector may not be the most reliable; an accurate classifier which is over- or under-confident will not be reliable.

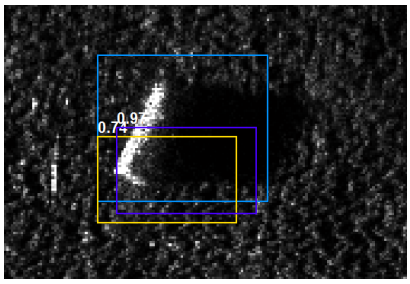


Fig. 1: Test-time detection of an object from an unseen class using detection uncertainties from two other object types. The large light blue rectangle represents ground truth information of the untrained cylinder object. The purple bounding box (confidence 0.97) of the uncertainty detector is shown. A false positive from a wedge shape detector (yellow, probability 0.74) also appears.

Ideally, if we have a detector which is not confident in the quality of the detections it returns, this should be reflected in its own output. In many situations we would arguably value reliable detections over accurate ones, especially if this allows the use of a robust method for selecting low-confidence detection regions and improving them. In surveillance or monitoring situations where the output of a detection algorithm is viewed directly by human operators, this can be resolved by asking the operator for a decision on the uncertain region. For autonomous systems where having a human in the loop would be impractical, regions containing low-confidence detections can be re-classified with a more accurate but more computationally intensive algorithm which would be too slow to apply throughout the entire scene. This uncertainty approach also helps when the penalty for false positives and false negatives is not equal; e.g. in mine-clearing operations, a false positive means time is wasted, while failing to detect a mine can have severe consequences. Accepting some level of ambiguity in initial detections is, we argue, an acceptable tradeoff.

Many classification algorithms use a hyperplane to separate one class from another. Support Vector Machine (SVM) classifiers assign more confident outputs to test samples further away from the hyperplane, but this results in classification predictions which are extremely *over*confident [4]. When

evaluated on a feature vector $\mathbf{x}$ representing a region a SVM will return a score $f(\mathbf{x})$; from this the sign $\text{sgn}(f(\mathbf{x}))$ gives the class label. N classification categories are used, $k = 1, \ldots, N$, and the score can be converted to a probabilistic prediction $p(C_k = 1|\mathbf{x})$ using a sigmoid function:

$$p(C_k|\mathbf{x}) = \frac{1}{1 + \exp(-2f(\mathbf{x}))}\,. \tag{1}$$

In various vision applications, the probability distribution has been shown to be biased towards 0 and 1 [4], [5]. In situations where multiple classifications are performed, however, samples for which no confident classification can be made have a higher uncertainty or entropy measure. As we show, this may indicate the presence of a class not present in the training data. Figure 1 shows detection of a cylinder (blue ground truth) at test time. We have not trained a cylinder classifier and thus this is an example of an anomalous object. It is detected using an uncertainty detector relying on the output of two trained classifiers (wedge and truncated cone shapes). This is shown in purple, and a false positive from the wedge detector is included too. Test samples distant from the clusters formed by training samples may also be an indication of the presence of an object class which is present at test time but was not included at training time. Related work follows in Section II. Next, in Section III we describe the classification method for individual detectors, followed by the uncertainty detector. We then describe the dataset and present and analyse results in Section IV. Conclusions are given in Section V.

II. Related Work

A. Detection with Uncertainty

We use a sliding-window approach to detection, common in the computer vision domain. In a traffic sign detection application, Grimmet *et al.* proposed the use of entropy information to identify sample windows which, after a classification stage, had high uncertainty [2]. This approach thus produced an *introspective* detector. Support vectors were compared to Gaussian Process Classifiers (GPCs) and Logitboost-based classifiers. For an application with sufficient training data to ensure all detectors were discriminative enough to be usable (i.e. precision, recall and F_1-score were all > 0.9), GP classifiers were found to have the highest average entropy when presented with samples of unknown classes at test time.

Gaussian Process classifiers produce probabilistic classifications directly from the best-fitting distribution using an activation function [6]. However, if probabilistic classifications are required from SVMs and Adaboost-based classifiers, these must be produced by converting the classifier score using an activation function. This 'squashes' all values far from the hyperplane into $[0, 1]$; (1) is an example. Platt described an improved method for fitting a parameterised sigmoid using a holdout data set [7]. Niculescu-Mizil and Rich describe isotonic regression (IR). This is a similar method, essentially a lookup table for converting classifier scores into a nondecreasing $[0, 1]$-valued distribution [8].

Blair *et al.* used a similar approach to [2] for a single-class pedestrian detection problem [5]. SVMs, GPs and Adaboost-based classifiers were applied following a robust gradient- and colour-based feature extraction stage. They showed that SVMs with Platt- or IR-based probabilistic classification were as reliable as GP-based classifiers for that application, while requiring considerably reduced processing time. This approach has also been shown to perform well in other remote sensing classification tasks [9].

B. Detection in SAS imagery

A common approach to object detection in SAS imagery is to use manual feature extraction to pick out features characterised by highlights and shadows in the generated imagery [1], [10]. Williams [1] has made a comprehensive study of many existing SAS datasets involving multiple different object types. He constructs a multi-stage detector comprising a shadow detector, a stage for removing seabed ripples (as they have a similar highlight-shadow pattern to underwater objects, they are a major source of false positives) and an highlight strength estimation stage. However, his work concentrates on automatic detection of all objects and does not take into account situations when we may wish to discriminate between different object classes, or locate and classify the presence of a previously unseen class.

Identification of previously unseen objects is a form of anomaly detection. Here we borrow a definition from Loy, who classifies anomalies in video data as "observed behaviour which is absent or rarely present in the training data" [11]. This can be subdivided into three types; events or objects which are strongly different from the training data, those which are ambiguous or rarely present in training, or those with weak visual evidence. In this case we define a new class as something strongly different from the training data.

C. Operator Trust and Human Factors

Let us consider situations where an object classification will be used directly by a human operator, such as a military or security-oriented surveillance task. Work in the human factors domain has established the effect of trust of a human operator in an automated system. Disuse (under-use of automation) and misuse (overuse of automation) [12] are both problems in this case. A high rate of false alarms will often cause a system to be under-used. Similarly, algorithm detection failures on 'easy' examples will cause disuse of a system [13]. Dzindolet *et al.* showed that, when operators were assisted in a detection task by an algorithm, they were more likely to trust the algorithm's decision when given reasons why the automated detection might fail [13]. This helped the users to arrive at an appropriate degree of reliance. This was reinforced in a later study by Madhavan *et al.*, who note that when operators rely on a machine to complete a task, it is initially expected to perform nearly perfectly [14]. They also found that machine errors when performing simple tasks reduce reliance and cause humans to rely on their own perceptual abilities more. This must be balanced against innate human overconfidence in their

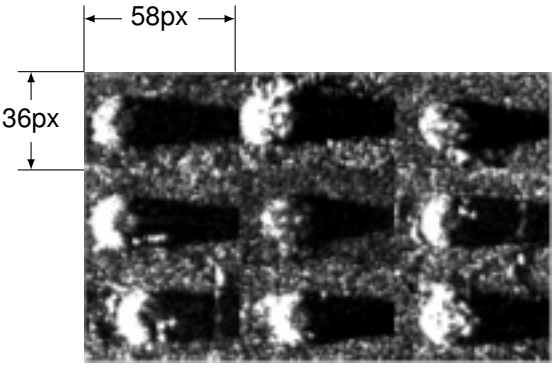


Fig. 2: Sample cone images in the training set, before normalisation.

own abilities, resulting in reduced algorithm-assisted overall performance. Thus, in the absence of 'perfect' detection algorithms, here we argue that providing information on overall algorithm reliability and reliable confidence measures on a per-detection basis, may improve operator trust in situations where an existing algorithm may be disused.

D. Contributions

In this work we describe the application of existing classification approaches to a challenging SAS data set where limited training data is available, and demonstrate the ability to identify regions where anomalous or previously unseen object classes are present. In doing so we extend the work of Grimmet *et al.* [2] and Blair *et al.* [5] on uncertain classifications and Niculescu-Mizil and Rich [8] to reliability, and apply it to multiple object classes.

III. Method

Given a dataset of training and testing images, the classification algorithms we describe in this section perform three separate tasks:

- detect locations of cones in the test dataset & provide confidence output
- detect locations of wedges in the test dataset & provide confidence output
- use uncertainty information (classifier disagreement) to identify image regions where an unseen class may be present. In this case we use cone and wedge detectors to flag areas where cylinders may possibly be located.

A. SVM Training

SVMs evaluate a feature vector $\mathbf{x}$ with a weight vector $\mathbf{w}$.

$$f(\mathbf{x}) = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}, \mathbf{w}_i) + b \tag{2}$$

This can be obtained with the linear kernel:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j^T \tag{3}$$

An alternative, the radial basis function (RBF) kernel, can also be used. This is generally more complex to evaluate but can have greater discriminative ability, as it can represent non-linear decision boundaries:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\right) \tag{4}$$

SVM-RBF and SVM-Linear classifiers were trained in a 1-vs-1 manner, on truncated cones, wedges and backgrounds using a 58×36 pixel window size and a window stride of $1/4$ of the window size. Each window was extracted then feature vectors were generated by normalising each one so that they had mean 0 and standard deviation of 1. No further preprocessing was performed. Three rounds of hard training were performed, where the classifier was evaluated on all the training data and retrained on failing samples. As the training images contained examples of three object types, for any image containing the "unknown" cylinder shapes, the windows which overlapped the cylinder area were removed, ensuring that cylinder shapes were completely unseen throughout the training process.

The LIBSVM library[1] was used throughout, and a GPU-accelerated implementation of the RBF kernel[2] was used to speed up training and testing.

B. Probabilistic Classification

The SVM output is reinterpreted probabilistically, using the fitting algorithm described by Platt [7]. This was used to obtain the a and b constants in (5):

$$p(C_k|\mathbf{x}) = \frac{1}{1 + \exp(-af(\mathbf{x}) + b)} \tag{5}$$

The holdout set used to learn the constants was obtained from a 4:1 split of the training data.

C. GPC Training

Gaussian Process Classifiers (GPCs) allow probabilistic prediction $P(y = +1|\mathbf{x}_*)$ of a new data sample $\mathbf{x}_*$, given a set of training data $\mathbf{x}$ and corresponding labels $y \in \{0, 1\}$. This is a two stage process. First a latent set of functions $f(\mathbf{x})$ is defined. This is assumed to have a Gaussian distribution, and thus can be described by its mean $\mu(\mathbf{x}) = 0$ and covariance function $k(\mathbf{x}, \mathbf{x}*)$. For a set of training and test points $(\mathbf{x}, \mathbf{x}_*)$, this is formulated:

$$\begin{bmatrix} \mathbf{y} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(x,x) & K(x,x_*) \\ K(x_*,x) & K(x_*,x_*) \end{bmatrix}\right). \tag{6}$$

The best-fitting distribution $\mathbf{f}_*$ is found, and this is bounded using an *activation function* with range $[0, 1]$ such as the logistic mapping function: $\sigma(x) = 1/(1 + e^{-f(x)})$. The mean and covariance of $\mathbf{f}_*$ can then be used to find:

$$p(y = +1|\mathbf{x}, y, \mathbf{x}_*) = \int \sigma(\mathbf{f}_*)\, p(\mathbf{f}_*|\mathbf{x}, y, \mathbf{x}_*) d\mathbf{f}_*\,. \tag{7}$$

This operation is $\mathcal{O}(n^2)$ for test cases and produces a probabilistic prediction value or 'soft' classification [15]. As GPCs

[1] www.csie.ntu.edu.tw/~cjlin/libsvm/

[2] wmii.uwm.edu.pl/~ksopyla/projects/cusvm-for-cuda-5-0-and-matlab-x64/

perform probabilistic classification directly, the extra conversion step in (5) is not needed. In a similar manner to SVMs, the covariance matrix when using linear kernels is:

$$k_{ij}(x) = \sigma_0^2 + \mathbf{x_i} \cdot \mathbf{x_j}^T . \quad (8)$$

We also trained with squared error kernels but these proved to be unable to discriminate between classes when training; results are therefore not included here. The GPML library[3] was used throughout.

D. Entropy

Uncertainty or entropy H can be measured using:

$$H = -\sum_{k=1}^{M} [p(C_k|\mathbf{x}) \log_M (p(C_k|\mathbf{x}))] \quad (9)$$

It operates on the probabilistic classifications obtained by (1), (5) or (6). This is normally expressed using base-2 logarithms [16, Ch.2]. Here we use base-M logarithms so that, for M discrete classes, H is bounded between 0 and 1. A larger H represents greater uncertainty in a classification and is maximised when $p(C_k|\mathbf{x}) = 0.5$ for all $k \in M$ classes. Entropy is therefore greatest when all individual classifiers are most unable to provide a confident prediction.

IV. Results

We measure classifier performance on two datasets in terms of misclassification rate and reliability in §IV-A. Following this, in §IV-B we use this information to evaluate our uncertainty detector.

Comprehensive synthetic aperture sonar datasets were gathered by NATO and first described in the literature by Groen *et al.* in 2009 [10]. MUSCLE SAS was used to gather Colossus 2 and Catharsis 2 datasets, of which a portion is used here. These datasets contain a number of simple shapes lying on the seabed and are used to test classifier performace at detecting man-made objects. Subsets from the Colossus 2 B and D sets (comprising 132 large 4000×1000 seabed images in total) were used for training. For testing we use subsets of the Colossus 2 C and Catharsis 2 image sets, with 69 and 168 images respectively. The images contain objects labelled as wedges, truncated cones, and cylinders, in a variety of different seabed conditions: flat and sandy, rippled or muddy, or covered in seagrass. Some example training cone shapes are shown in Figure 2.

A. Detection of Cones and Wedges

Detection performance for classifying cones and wedges is given in Table I for the Colossus 2 C set and in Table II for the Catharsis 2 set. We show true positives (TP), false negatives and positives (FN, FP), precision $p = TP/(TP + FP)$, recall ($r = TP/(TP + FN)$) and F_1-score:

$$F_1 = \frac{(\alpha + 1)rp}{r + \alpha p}, \alpha = 1 \quad (10)$$

[3]www.gaussianprocess.org/gpml/code

TABLE I: Test set classification performance for Colossus2C dataset. 61 images, seafloor complexity similar to training data.

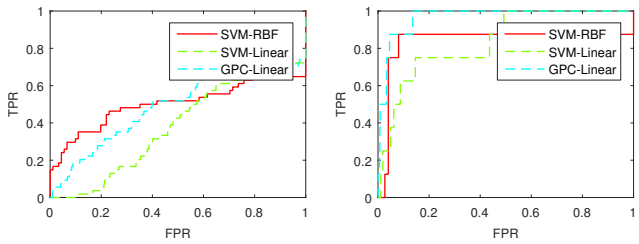
Classifier	Object	TP	FN	FP	p(%)	r(%)	F_1
SVM-RBF	cone	35	19	91	28	65	0.389
SVM-RBF	wedge	44	16	4655	0.93	73	0.018
SVM-Linear	cone	39	15	18879	0.21	72	0.004
SVM-Linear	wedge	0	60	203	0	0	0
GPC-Linear	cone	40	14	15569	0.26	74	0.004
GPC-Linear	wedge	54	6	15330	0.015	90	0.003

Each instance of the object in a separate view is counted independently.

We also give ROC curves for cones in Figure 3 and for wedges in Figure 4. From these, the RBF classifiers always outperform the linear ones. The wedge detectors also perform relatively poorly; cones (with their rotationally invariant appearance) are more consistently detected when compared to wedges. As expected, radial basis function SVMs also perform better than their linear equivalents. In some cases (e.g. Figure 4a) the linear SVM provides no discriminative ability. The limited performance for all detectors on the Catharsis dataset is due to the lower number of ground-truth detections.

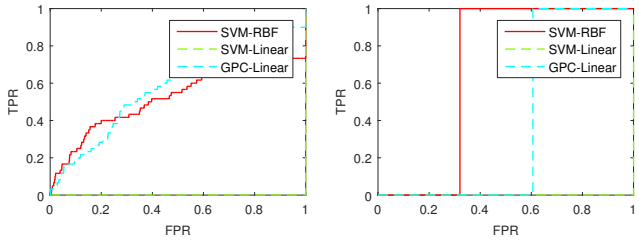
The linear GPC outperforms the linear SVM, but both provide an unacceptable number of false positives. The linear GPC does not outperform the non-linear kernel SVM, however, showing the limitations of a linear decision boundary. The major problem with all detection approaches is that a high number of false positives are produced, such as for the rippled seabed in Figure 5. These are (i) mainly present in Colossus 2 C and not in B and D – hence only in the testing data – and (ii) particularly a problem when the highlight and shadow size of the ripples is the same size as the cone and wedge features. See Section V for one possible approach to this problem. Table III shows a comparison of the AUC results from the subsets of each dataset that we use, against existing work from 2015 which covers the mine-like object detection over the complete datasets [1]. Our results do not improve on existing work, but (i) we are instead concerned with the problem of detector reliability and detection of novel classes and (ii) not explicitly setting out to build a replacement object detector. Williams uses a seabed ripple detector to remove false positive classifications as a middle stage, but does not discriminate between object classes [15].

In Figure 6 and Figure 7 we show reliability diagrams of all detection approaches. Here, a 'well-calibrated' or perfectly reliable detector would lie on the black line, i.e. of all the detections it predicts with 60% confidence, 60% will be evaluated as true. The RBF classifier is always more reliable than the linear version. The wedge detectors are unreliable, again due to the rotationally variant properties of the shape. However, all detectors are significantly overconfident and further post-processing of the results is required for any of these detectors to be considered *well-calibrated*. This is again the result of a high number of false positives dues to ripples.



(a) Cone Colossus2c (b) Cone Catharsis2

Fig. 3: ROC curves for the cone detector.



(a) Wedge Colossus2c (b) Wedge Catharsis2

Fig. 4: ROC curves for the wedge detector.

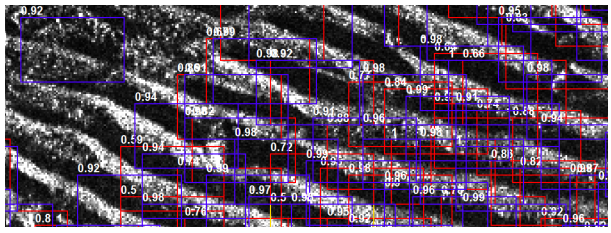
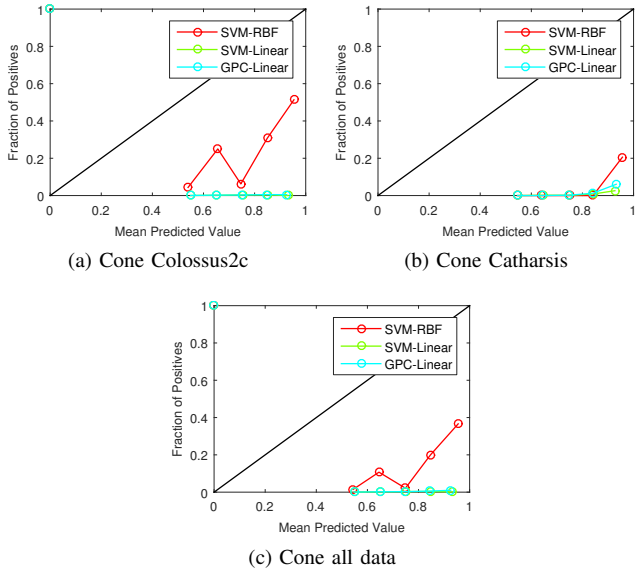


Fig. 5: Seabed ripples in the test set produce many false positives from cone and wedge detectors.



(a) Cone Colossus2c (b) Cone Catharsis

(c) Cone all data

Fig. 6: Reliability diagrams for the cone detector.

TABLE II: Test set classification performance for Catharsis2 dataset. 168 images, seafloor more complex than training data.

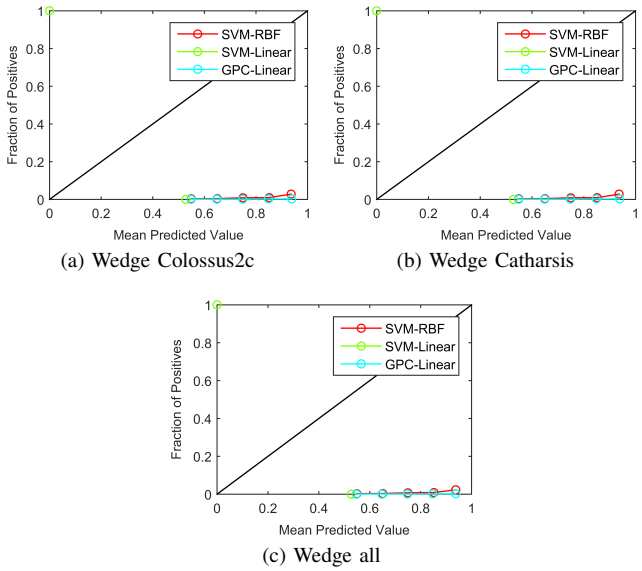
Classifier	Object	TP	FN	FP	p(%)	r(%)	F_1
SVM-RBF	cone	7	1	149	4.9	100	0.085
SVM-RBF	wedge	1	0	836	0.11	100	0.002
SVM-Linear	cone	8	0	3916	0.20	100	0.004
SVM-Linear	wedge	0	1	5	0	0	0
GPC-Linear	cone	8	0	3267	0.24	100	0.005
GPC-Linear	wedge	1	0	15330	0.007	100	.0001

TABLE III: Area Under Curve (AUC) performance of our detectors on the subsets of Colossus2C and Catharsis2, compared to other work on the entire datasets.

Dataset	Object	Classifier	AUC
Colossus2C	cone	SVM-RBF	0.502
	cone	SVM-Linear	0.383
	wedge	SVM-RBF	0.518
	wedge	SVM-Linear	0
	cone	GPC-Linear	0.493
	all	Williams [1]	0.803
Catharsis2	cone	SVM-RBF	0.836
	cone	SVM-Linear	0.837
	wedge	SVM-RBF	0.679
	wedge	SVM-Linear	0
	cone	GPC-Linear	0.966
	all	Williams [1]	0.960

B. Unknown Object Classes

We now consider capability for detection of previously unseen classes. Using the entropy information from each detector, we expect this to be maximised where the SVMs are 'uncertain' and have very little discriminative ability;



(a) Wedge Colossus2c (b) Wedge Catharsis

(c) Wedge all

Fig. 7: Reliability diagrams for the wedge detector.

TABLE IV: Test set classification performance: Using uncertainty to detect unseen cylinders in the Colossus2C dataset. Uncertain detections are those with $H > 0.9$ and not overlapping a known detected object.

Classifier	"TP"	"FN"	"FP"	p(%)	r(%)	F_1-score
SVM-RBF	12	15	17	41.4	44.4	0.42857
SVM-Linear	24	3	2804	0.84	88.9	0.01681

TABLE V: Test set classification performance: Using uncertainty to detect unseen cylinders in the Catharsis2 dataset. Uncertain detections are those with $H > 0.9$ and not overlapping a known detected object.

Classifier	"TP"	"FN"	"FP"	p(%)	r(%)	F_1-score
SVM-RBF	1	0	57	1.72	100	0.03389
SVM-Linear	1	0	2640	0.03	100	0.00057

i.e. where for both the detection probabilities are $p \sim 0.5$. Thus, for every image we threshold H from (9) at 0.9, and discard those bounding boxes which overlap with existing detections. Entropy thresholding removes many of the FPs caused by seabed ripples, leaving behind the unknown objects. See Table IV; in the "TP", "FN" and "FP" columns we list detections of ground-truth cylinder locations with a detector which has never been trained to detect them. With a SVM-RBF classifier, the recall rate of samples of an unknown class is 44%; arguably enough to signify to an operator or autonomous system that there is potentially an additional object class present which merits further investigation. We also express this in a ROC curve for Colossus2C in Figure 8. For the Catharsis set given in Table V, only one ground-truthed cylinder was present, and this was detected with numerous false positives. There are examples of this detector in Figure 1 and failure modes are shown in Figure 9, where the uncertainty detector also identifies a wedge missed by the wedge detector. Interestingly, the RBF-based uncertainty detector is less susceptible to false positives caused by the seabed ripples than the original object classifiers; we speculate that this is because each object detector responds differently to the same area of seabed.

V. Conclusion

We have described SVM- and GPC-based detectors for two object types in synthetic aperture sonar data. While these do not achieve state-of-the-art detection results, we are able to compute a a probabilistic prediction of detection performance then use the uncertainty arising from this to identify samples from a previously unseen class at test time. In the best case we can recall 44% of the samples of a previously unknown class. Where possible, we have included Gaussian Process-based detectors as comparison. The main issue with our approach is the high number of false positives found by the original detectors in rippled seabed areas; improved methods to tackle these are required. Recent work by Daniell *et al.* has shown that regional context classifiers can suppress false positives, at little computational cost [17]; this approach is being considered for future work.

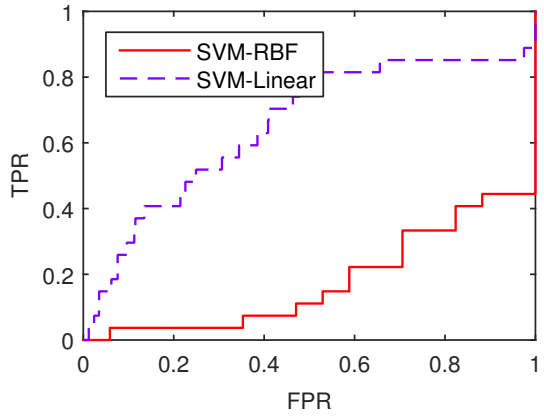
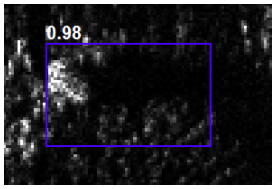
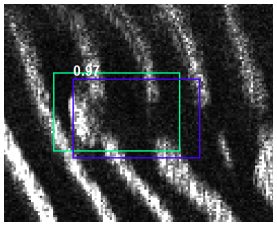


Fig. 8: ROC curves for unseen detection of cylinders using uncertainty in Colossus 2C.



(a) Non-mine object (false positive) identified with high confidence by the uncertainty detector.



(b) Wedge (green) in rippled (hard) area missed by wedge detector but identified by the uncertainty detector (blue).

Fig. 9: Failure modes of uncertainty detector.

Acknowledgements

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/J015180/1 and the MOD University Defence Research Collaboration in Signal Processing. The MUSCLE dataset was supplied by the Centre for Maritime Research and Exploration (CMRE).

References

[1] D. P. Williams, "Fast Target Detection in Synthetic Aperture Sonar Imagery: A New Algorithm and Large-Scale Performance Analysis," *IEEE Journal of Oceanic Engineering*, vol. 40, no. 1, pp. 71–92, Jan. 2015.

[2] H. Grimmett, R. Paul, R. Triebel, and I. Posner, "Knowing When We Don't Know: Introspective Classification for Mission-Critical Decision Making," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, May 2013.

[3] D. J. Hand, *Construction and Assessment of Classification Rules*. Wiley, 1997.

[4] R. Triebel, H. Grimmett, R. Paul, and I. Posner, "Driven Learning for Driving: How Introspection Improves Semantic Mapping," in *International Symposium on Robotics Research (ISRR)*, 2013, pp. 1–16.

[5] C. G. Blair, J. Thompson, and N. M. Robertson, "Introspective Classification for Pedestrian Detection," in *Sensor Signal Processing for Defence (SSPD 2014)*, Edinburgh, 2014.

[6] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. University Press Group Limited, 2006.

[7] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in large margin classifiers*, 1999.

[8] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *International Conference on Machine Learning*, no. 1999, 2005.

[9] H. Dehghan and H. Ghassemian, "Measurement of uncertainty by the entropy: application to the classification of MSS data," *International Journal of Remote Sensing*, vol. 27, no. 18, pp. 4005–4014, Sep. 2006.

[10] J. Groen, E. Coiras, and D. Williams, "Detection rate statistics in synthetic aperture sonar images," in *Proceedings of the 3rd International Conference and Exhibition on Underwater Acoustic Measurements*, no. June, 2009, pp. 21–26.

[11] C. C. Loy, T. Xiang, and S. Gong, "Detecting and discriminating behavioural anomalies," *Pattern Recognition*, vol. 44, no. 1, pp. 117–132, Jan. 2011.

[12] R. Parasuraman and V. Riley, "Humans and Automation: Use, Misuse, Disuse, Abuse," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 39, no. 2, pp. 230–253, Jun. 1997.

[13] M. T. Dzindolet, S. A. Peterson, R. A. Pomranky, L. G. Pierce, and H. P. Beck, "The role of trust in automation reliance," *International Journal of Human-Computer Studies*, vol. 58, no. 6, pp. 697–718, Jun. 2003.

[14] P. Madhavan, D. A. Wiegmann, and F. C. Lacson, "Automation failures on tasks easily performed by operators undermine trust in automated aids." *Human factors*, vol. 48, no. 2, pp. 241–56, Jan. 2006.

[15] C. K. I. Williams, "Gaussian Processes," in *The Handbook of Brain Theory and Neural Networks*, 2nd ed., M. A. Arbib, Ed. Cambridge: MIT Press, 2002.

[16] T. Cover and J. Thomas, *Elements of Information Theory 2nd Edition*. Wiley-Interscience, 2006.

[17] O. Daniell, S. Reed, J. Vazquez, and A. Frau, "Reducing False Alarms in Automated Target Recognition using Local Sea-floor Characteristics," in *Sensor Signal Processing for Defence (SSPD 2014)*, 2014, pp. 21–25.