Naval Research Laboratory
Washington, DC 20375-5320

NRL/7350/MR—2023/3

Automated Detection of Whale Signatures in PAN and MSI Imagery Using Feature Vectors Combined With Neural Net Algorithms

CHRISTOPHER WACKERMAN

Seafloor Sciences Branch
Ocean Sciences Division

July 13, 2023

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
13-07-2023	NRL Memorandum Report	01/04/2021 – 31/12/2022

4. TITLE AND SUBTITLE

Automated Detection of Whale Signatures in PAN and MSI Imagery Using Feature Vectors Combined with Neural Net Algorithms

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
0602435N

6. AUTHOR(S)

Christopher Wackerman

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
1M20

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
1005 Balch Blvd.
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/7350/MR--2023/3

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
One Liberty Center
875 N. Randolph Street
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A: Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Description of a tool to detect whale signatures in commercial panchromatic and multi-spectral images using a combined feature vector and neural net detection algoritm.

15. SUBJECT TERMS

Whale signature detection
Multi-spectral imagery
Panchromatic imagery

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Christopher Wackerman
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U	U	49	(228) 688-5354

This page intentionally left blank.

Automated Detection of Whale Signatures in PAN and MSI Imagery Using Feature Vectors Combined With Neural Net Algorithms

1.0 Summary

In this paper we present an automated detection algorithm for whale signatures in optical satellite remote sensing imagery that is an attempt to design an approach that will perform better on smaller test data sets than the traditional deep learning CNN approaches. The detection algorithm combines a feature vector with either an optimal linear combination of feature elements or a neural net to decide whether the feature vector comes from a whale signature or not. The feature vector used in this study is a combination of normalized statistics, statistics designed to determine if the image values within some target box are statistically different than those in the local background, and a set of metrics designed to measure image properties unique to whale signatures in panchromatic imagery. A test set was extracted from image data spanning WorldView-2, WorldView-3, GeoEye-1, Quickbird-2, and Pleiades imagery using a combination of manually extracted whale signatures by Drs. Hannah Cubaynes and Peter Fretwell of the British Antarctic Survey and signatures extracted using an anomaly filter developed in this study. From these a total of 730 whale signatures and 9300 background signatures were extracted to provide training/testing data.

To develop the algorithms an optimal linear combination algorithm was developed based on linear discriminant theory, and a neural net was designed to be used with the training data. Both of these were then run on the dataset using 10 trials that randomly divided the data into training and testing data. Two metrics were used to determine performance of the algorithms: a plot of probability of detection (PD) versus probability of false alarm (PFA) that is normally referred to as a Receiver Operating Characteristic Curve (ROCC), and the expected number of false alarms generated over a region the size of a WorldView image tile (NFA/Tile). The two algorithms were run on various combinations of image types and normalization approaches. In addition, the detection algorithms were run on full imagery to compare with the training/testing set results.

Finally we compared using the feature vectors with using the image values directly into a neural net, the latter meant to simulate a simple CNN, in order to determine if there was any improvement in performance using feature vectors instead of image values when the training set size is small (as is the case here).

Conclusions from this study are:

- Using a feature vector approach performed better than using image values directly into either the optimal linear combination or neural net algorithms, giving an indication that for small training sets a feature-based approach will provide improved performance over standard CNN approaches;

- Our results were either better or equivalent to published results that used significantly larger training sets and significantly more complex CNN neural nets, again giving an

Manuscript approved July 11, 2023.

indication that for small training sets a feature-based approach can provide improved or equivalent performance over traditional CNN approaches.

- Using a neural net to determine which class a feature vector belongs to performed better than using an optimal linear combination, and both ran in about the same time on an image;

- Using multi-spectral imagery gave better performance than using panchromatic imagery, even with the additional image metrics applied to the higher resolution Pan imagery;

- Using WorldView 8-band multi-spectral imagery the proposed algorithm generates a probability of detection of ~ 97% at a probability of false alarm of ~1% with 10-100 false alarms within a WorldView tile. Using 4-band multi-spectral imagery from all sensors, the proposed algorithm generates a probability of detection of ~ 94% with a probability of false alarm of ~ 1% with ~200 false alarms per WorldView tile.

- The detection algorithm takes from 500 to 1000 seconds to run on a WorldView Tile. For a standard WorldView collection that might contain 30 tiles, the time to process the entire collection would be 4-8 hours.

- WorldView 8-band multi-spectral Ortho-rectified imagery provides the best performance, most probably due to the Dynamic Radiometric Adjustment that is applied to the image to enhance contrast.

Details that support these conclusions now follow. A summary of ROCC and NFA/Tile results over a range of image types and algorithms analyzed in this study is shown in Figure S1. The results fall into four main groups:

- Using WorldView multi-spectral imagery (MSI) Ortho-rectified or Basic with the neural net algorithm are the best, top, results with NFA/Tile=12/101 and a PD~97% for PFA of 1% (0.01). Performance that is well within operational capability, and they stand out from the rest of the image combinations.

- The next group consists of (a) all MSI images using 4 bands with the neural net algorithm; (b) WorldView Ortho panchromatic (PAN) imagery with the neural net algorithm; and (c) Worldview Ortho-rectified MSI with the optimal linear algorithms. These all generate NFA/Tile ~ 190 (from 179 to 206) with a PD~94% for PFA=1%. These false alarms could be easily reviewed by a user in an operational setting so we are still within operational utility.

- The next group consists of (a) WorldView Orth-rectifiedo PAN imagery with the optimal linear algorithms; (b) WorldView Basic PAN with neural net; and (c) All PAN imagery with the neural net algorithm. NFA/Tile ~ 600 (from 511 to 747) with a PD~90% for PFA=1%. Note that for this group the NFA/Tile is starting to get too high for a user to easily review in an operational setting, and a PD of 90% may be too low.

- The final group consists of all optimal linear algorithms using (a) all PAN imagery; (b) WorldView PAN Basic; (c) all MSI imagery; and (d) WorldView Basic MSI. The NFA/Tile ~ 1400 (from 1082 to 1827 with the excepthon of WorldView Basic MSI that fluctuates wildly) and with a PD~86% for PFA=1%. Note that for this group the NFA/Tile is more than we would want a user to have to review in an operational setting.

Clearly we want to stay within the first two groups, which means that we want to use either : (a) only WorldView MSI imagery (Ortho or Basic) or PAN Ortho with the neural net algorithm; or (b) use any available satellite sensor system, in which case we want to use MSI with the neural net algorithm. These results also show that neural net is better than optimal linear (4/5 of the top two groups are neural net and all of the last group are optimal linear) and panchromatic is better than multi-spectral (4/5 of the top two groups are MSI and 5/7 of the bottom two groups are PAN).

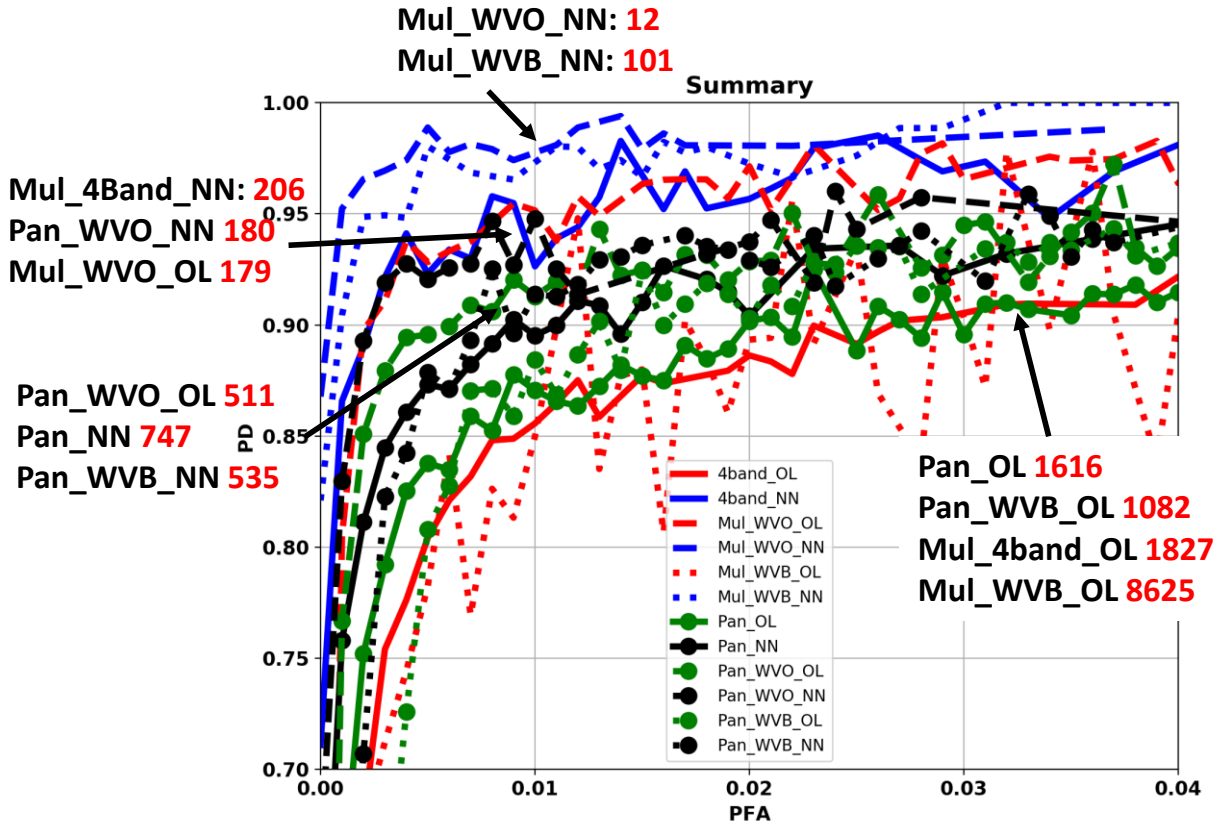


Figure S1: Summary of the ROCCs for the various combinations. For all the MSI results in this plot, no normalization was used. For all the Pan results, they were normalized by the mean within the target box. The notations around the plot show what images/algorithms each ROCC corresponds to (as does the legend) along with a red number that is the average NFA/Tile for a PD=90%.

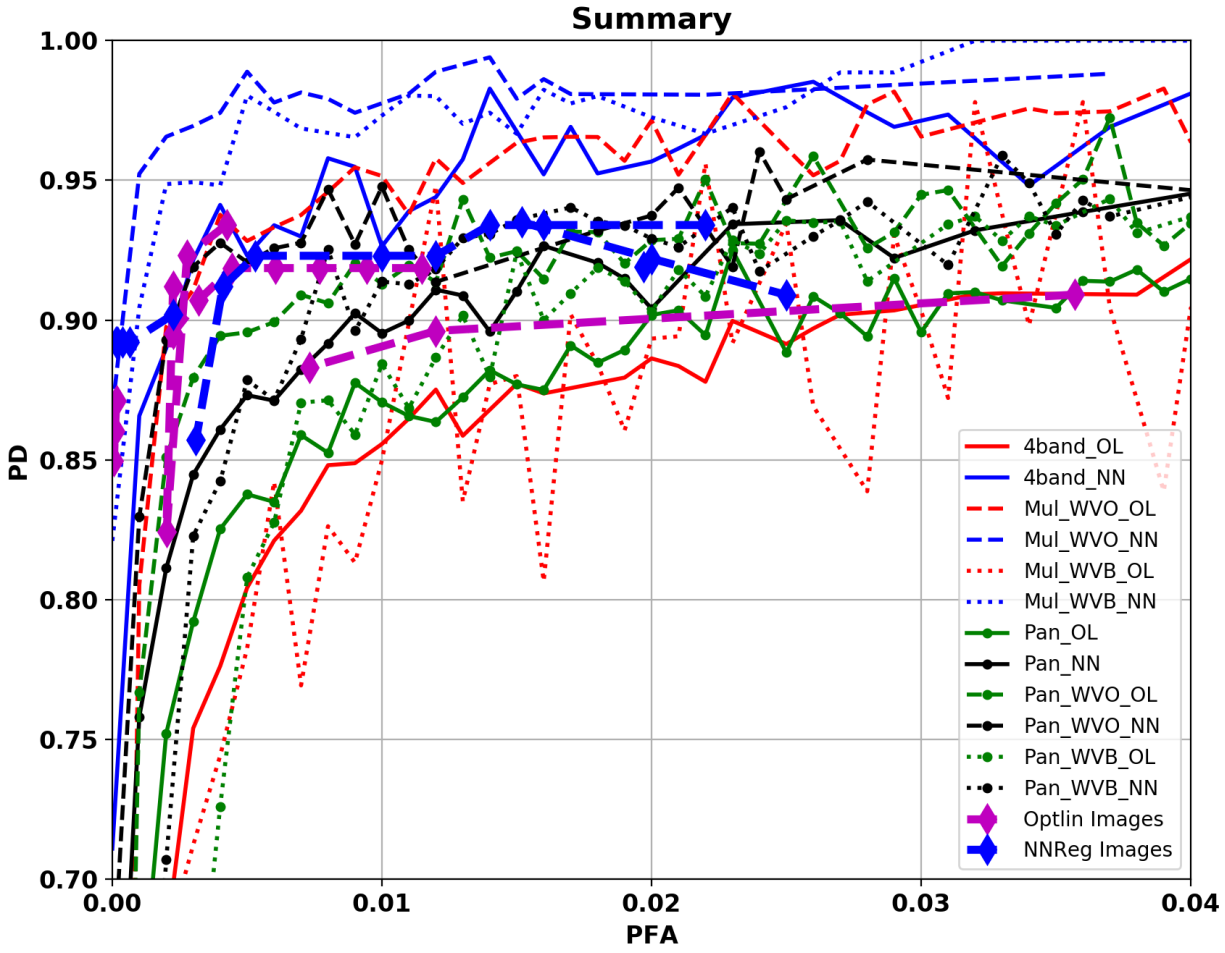


Figure S2: Same results as in Figure S1 (but drawn with thinner lines and smaller symbols) with the results from running the detection algorithm on full imagery added. Purple diamonds are full image results using the optimal linear algorithm. Blue diamonds are full image results using the neural net regression algorithm. The image results come from four images: Valdes 2012, Valdes 2014, Valdes 2016, and Witsand 2009 and were run over the tiles which contained the manually-derived whale signatures.

To check these results, Figure S2 shows the same data, but with the results from running the detection algorithms on Pan images from four image cases. All but one of the images generated PD ~93% for PFA in the range from 0.005 to 0.02 which spans the regions of group 2 and group 3 as discussed above. Note that these groups span the test set results for WorldView Ortho Pan images, either with neural net or optimal linear algorithms, and all Pan images with neural net algorithm, so the image results appear consistent with the test set results. The remaining image generated PD~ 90% for this same range using the optimal linear algorithm and is consistent with the lower portion of group 3; so again consistent but at the lower end. Thus the full image PAN results appear consistent with the PAN test set results. To date we have only looked at PAN images; future work will apply the algorithms to full MSI Ortho images to see if we can generate results consistent with the better Mul_WVO_NN curve in figures S1 and S2.

The training set in this study was derived from manual analysis without any *in situ* ground truth, so there is probably a subset of our "whale" image chips that are in fact not whales but small boats or judiciously placed breaking waves. This means that our reported PFA values are probably too low since we have put actual false alarms into the "whales" category. However without any ground truth it is difficult to determine how far off our PFA values may be.

The results shown here are specific to our test set, the set of features we used in the feature vector, and the specific parameters for the neural net. In future work we will determine if a more complex neural net will significantly improve performance, and we will continue to search for better features to use for whale signature detection. As mentioned above we also will apply the detection algorithms to full MSI Ortho images to determine if we can get image results consistent with the test set ROCCs.

2.0 Introduction

There has been a large body of work recently examining the capability of high-resolution satellite imagery as a tool for marine mammal monitoring and counting; including penguins (Barbra-Meyers et al., 2007; Fretwell and Trathan, 2009; Fretwell et al., 2012), seals (LaFue et al., 2011; McMahon et al., 2014), polar bears (Stapleton et al., 2014) and whales (Abileah, 2001; Abileah, 2002; Fretwell et al., 2014; Cubaynes et al., 2019; Borowicz et al., 2019; Guirado et al., 2019, Fretwell et al., 2019; Cubaynes et al., 2020; Clarke et al., 2021). The focus of this report is on whale signatures, and specifically to analyze the current operational feasibility of using satellite remote sensing to monitor whale activity when we have only a small number of whale signatures to use for any supervised classification algorithm.

Previous work has demonstrated that whales that are on or near the surface can be visually detected in high resolution (less than ~50cm) satellite imagery (Fretwell et al. 2014; Cubaynes et al. 2019; Fretwell et al., 2019; Clarke et al., 2021) and can often be separated from the surrounding water based on their spectral signature in Multiple-Spectral Imagery (MSI) (Abileah, 2001; Abileah, 2002; Fretwell et al., 2014; Cubanynes et al., 2019). Due to the large areas that have to be searched it has been recognized that for operational feasibility we need automated algorithms for whale signature detection, and results for automated approaches have been shown using standard supervised and unsupervised algorithms based on spectral signatures (Fretwell et al., 2014) and using convolutional neural nets (CNN) applied to the MSI (Borowicz et al., 2019; Guirado et al., 2019). (Fretwell et al., 2014) showed results from satellite imagery with a resolution of 50 cm or better. Both (Borowicz et al., 2019) and (Guirado et al., 2019) used a combination of aerial and satellite imagery (mainly aerial) to get to large enough training sets to robustly train their CNN, then (Borowicz et al., 2019) did testing only on 31cm resolution satellite images whereas (Guirado et al., 2019) did testing on both aerial and satellite images. The CNN in both publications used a large number of layers, and the standard approach of training all but the last layers using huge databases of images that are on the web (and do not contain whales), then re-training the last layers using their whale images. In both publications, the total number of whale signatures from aerial images was around 700, with a few tens of whale signatures from satellite images.

Most of the references above note that there is a lack of satellite imagery whale signature examples compared to non-whale examples, and that this is particularly a problem for deep learning (particularly CNN) approaches since they often required training sets on the order of 100,000s to millions of examples or more. One way too address this issue is to look at the feasibility of using automated algorithms for satellite whale signature detection that generally do not require large training sets; in this paper we specifically look at feature-vector based algorithms. However in some sense we have just kicked the problem down the road slightly; CNNs require large training sets, but once you have the training set the CNN are automatically generated and have been shown to work extremely well on many applications. For feature-vector based algorithms, we do not need the large training sets, but we have to come up with "good" set of features designed for this specific applications, otherwise they will not work well. So we are trading numbers of examples for designing good features. In addition, we do not get completely out of the need for whale signature examples; the algorithms discussed below still

require training on known signatures. However the results in this paper indicate that feature-vector approaches can provide improved performance with much smaller training set sizes.

This paper will report on automated detection results from two types of algorithms that use feature vectors to differentiate between whale and non-whale signatures: (1) an algorithm that applies an optimal linear combination of feature vector entries; and (2) and an algorithm that uses a neural net to determine which class a feature vector came from. We will determine performance of the algorithms using two metrics: (1) a plot of the probability of a false alarm (PFA) versus the probability of detection (PD) which is referred to as a Receiver Operating Characteristic Curve (ROCC); and (2) the total number of false alarms per image for some set PD value, where we will define an image as one tile from the WorldView sensor system (one tile is usually what is viewed as an "image" from the collected data using the WorldView sensor). To calculate the number of false alarms per tile (NFA/Tile) we first calculate how many applications of the detector would be used within a tile, then multiple this by the PFA for the given PD from the ROCC. For the results in this paper we calculate NFA/Tile for a PD=90%. For comparison, we will also look at performance of these two algorithms when the feature vector is just the image values themselves. This will give us a sense of any advantage that using features instead of image values may bring for small numbers of training data.

To allow us to compare to previous results we derive values of PD and PFA from the references above. (Fretwell et al., 2014) showed that the best performance was achieved by simply thresholding the spectral data; he reports a probability of detection (PD) of 85% and a probability of false alarm (PFA) of 24% compared to a manual interpretation of the imagery (see Table 1 in the article). For CNNs however, results are usually reported in slightly different terms than PD and PFA so we have made a conversion from both publications as follows. From (Borowicz et al., 2019) we used the confusion matrix results in Figure 4 from that paper that showed results using a total of 32 whale signatures and 1390 non-whale signatures. From these, the CNN detected all 32 of the whale signatures as whales and 87 of the non-whale signatures as whales. Due to the small number of whale signatures in these results, we estimated the PD as being in the range of 97%-100% (the lower bound coming from the fact that the next lower PD possible is 31/32 = 96.9% and the data set is not large enough to determine where we are within that range) and the PFA as being 87/1390 = 6.3%. For (Guirado et al., 2019) they report PFA directly as being from 1% - 2.25% (see the Results section, "Whales presence detection model (step-1) validation" paragraph) and they report that 62 whale signatures were reported as whale signatures out of a total of 84 signatures that were manually extracted from the imagery (see Results section, "Whale counting (step-2) model validation" paragraph) which gives a PD in the range of 73%-74%.

This report is divided as follows. Section 3 discusses the data set that we will use and how it was generated. Section 4 shows what features we selected for this analysis to use in the feature vector. Section 5 derives the optimal linear combination of features that we will use as one of the algorithms. Section 6 describes the neural net that we used for the other algorithm. Section 7 shows the results of applying the two algorithms to the data set and provides conclusions as to the performance of the approaches. Finally, Section 8 presents a summary and conclusions.

3. Satellite Imagery Data Set.

The satellite imagery used in this analysis were the following images:

- Auckland 2006 – this is a QuickBird-2 image collected 12 August 2006 and had 4 spectral bands (B,G,R,NIR) and a panchromatic band. Resolution is 2.6 meters for MSI and 0.66 meters for PAN;
- Witsand 2009 – this is a GeoEye-1 image collected 09 August 2009 and had 4 spectral bands (B,G,R,NIR) and a panchromatic band. Resolution is 1.7 meters for MSI and 0.43 meters for Pan;
- Auckland 2011 – this was a WorldView-2 image collected 27 August 2011 and had 8 spectral bands (C,B,G,Y,R,Re,NIR,NIR2) and a panchromatic band. Resolution is 1.9 meters for MSI and 0.47 meters for PAN;
- Valdes 2012 – this is a a WorldView-2 image collected 19 September 2012 and had 8 spectral bands (C,B,G,Y,R,Re,NIR,NIR2) and a panchromatic band. Resolution is 2.1 meters for MSI and 0.52 meters for PAN;
- Valdes 2014 – this is a a WorldView-3 image collected 16 October 2014 and had 8 spectral bands (C,B,G,Y,R,Re,NIR,NIR2) and a panchromatic band. Resolution is 1.4 meters for MSI and 0.36 meters for PAN;
- Valdes 2016 – this is a a WorldView-2 image collected 23 September 2016 and had 8 spectral bands (C,B,G,Y,R,Re,NIR,NIR2) and a panchromatic band. Resolution is 2.2 meters for MSI and 0.54 meters for PAN;
- Cape Code 2021 – this is a Pleiades image collected 11 March 2021 and has 4 spectral bands (B,G,R,NIR) and no panchromatic channel. Resolution is 0.73 meters for MSI

The name of the image is the location where it was collected, the date is the date of collection, and the bands correspond to C=Coastal Band, B=Blue, G=green, Y=yellow, R=red, RE=red edge NIR= near-IR, NIR2= the second NIR band on WorldView. There are a number of imagery products that can be downloaded from the DigitialGlobe Archive for these images. For the WorldView images one can get Ortho-rectified data, Basic data, or special products (such as pan-sharpened spectral images). For the GeoEye and QuickBird data there are no Ortho-rectified data available; just Basic and pan-sharpened. For WorldView data, the Ortho data is in what is call Dynamic Radiometric Adjustment (DRA) modes which means that there is an unknown (at least to us) modification to the spectral values performed on each image, which changes from image to image and spatially throughout one image, that is meant to improve the contrast of the imagery. We downloaded the Ortho MSI, Basic MSI, and Pan images (except for the Pleiades image) for each of the data collections above.

Drs. Hannah Cubaynes and Peter Fretwell performed a manual analysis of all of the images above, except for Cape Code 2021, and visually extracted image chips of what they determined were possible whale signatures (a subset of this analysis can be found in (Cubaynes et al., 2019)). This set is the bulk of the labeled image chips that were used in this analysis. However we also needed some method of searching very large images for possible whale signatures, particularly for the Cape Cod 2021 Pleiades image, but also to see if other signatures were present in the other images. To do this we developed an anomaly filter that would find all the image chips that were statistically different than the local background around the chip and

present them to a user for review. The user would then manually save those that visually were whale signatures.

The anomaly filter moves a small window through the image, and locates image chips which are statistically different then the local background. We set up a series of nested windows as shown in Figure 1, where the inner window is called the target box and will be the image chip that we are testing for whether it contains a target signature. We will calculate statistics within this window to test if they are different than the local background statistics. There is a guard ring around the signal window that allows us to have a target signature bleed out of the signal window but not affect the background statistics. Then there is a background ring that surrounds the guard ring; this is the region where we calculate the local background statistics. Our underlying assumption will be that we have only one target within the set of nested windows.

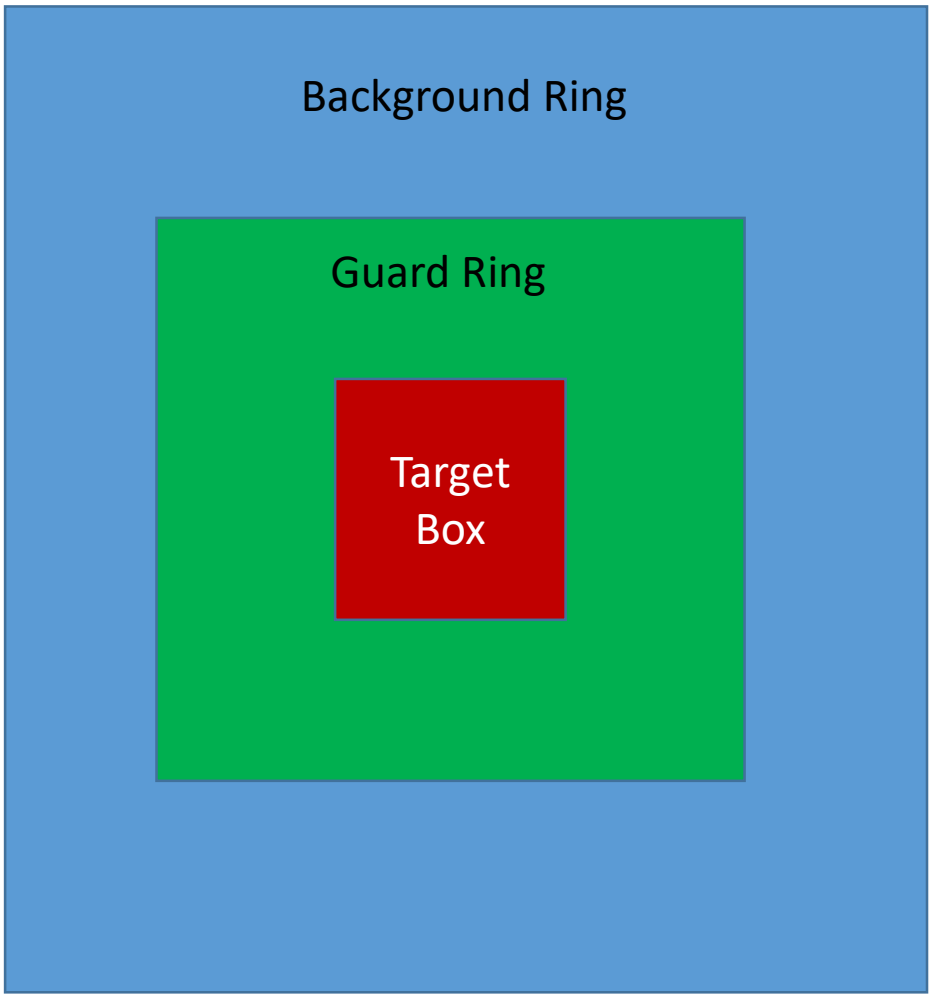


Figure 1: Schematic showing the placement of the nested windows being used to calculate local statistics for the anomaly filter.

This set of nested windows is moved through the image, jumping by some number of lines/columns each time. Because of this discrete jumping it is possible that the target signature may go across target signature boxes, and thus not provide the best statistics for characterizing the target. So for each placement of the nested boxes, we calculate the location of the brightest pixel in the guard box and shift everything so that the brightest pixel is in the middle of the target box. For multi-spectral data, we calculate the brightest pixel by first scaling the red, blue, and green channels separately to set their minimum value to 0 and their maximum to 1, then sum the three scaled images and find the maximum value. This is an attempt to locate the brightest pixel that would be visible to the eye in a colored image. We are assuming that if there is a target signature within the guard box then the brightest pixel will be near to the middle of the signature Note that this means that there may be portions of the image that never get covered by a target box; we are only examining one target box location within any given guard box. This should not cause issues however if our assumption of one target per nested windows is correct.

Ideally, the target box should be set to the expected size of the target signature we are trying to find. Then the background ring should be set to the largest size possible and still not contain another target signature; i.e. the background ring size is set by the expected distance between target signatures in the image.

For each placement of the nested boxes, we want to determine if the pixels within the target box are statistically different than the pixels in the background ring. We will assume that the image pixel values are independent, identically distributed, Gaussian random variables. For any given placement of the nested boxes, we will assume that the mean and standard deviation from the background ring represent the "true" mean and standard deviation of the background Gaussian random variable. The question then becomes do the pixels in the target box have the same Gaussian distribution. We will estimate the mean and variance of each region using the well known, unbiased, estimates of the mean and variance:

$$\hat{m} = \frac{1}{N}\sum_{i=1}^{N} X_i \quad (1)$$

$$\hat{v} = \frac{1}{N-1}\sum_{i=1}^{N} (X_i - \hat{m})^2 \quad (2)$$

where N is the number of pixels in the region where we are calculating the statistics and X_i are the individual image pixel values which we are assuming are independent samples of a Gaussian distributed random variable. If the underlying random variables have mean m and variance σ^2, then it is well known that

$$E[\hat{m}] = m, \;\; E[\hat{v}] = \sigma^2 \quad (3)$$

and

$$var[\hat{m}] = \frac{\sigma^2}{N}, \;\; var[\hat{v}] = \frac{2\sigma^4}{N-1} \quad (4)$$

and that $\hat{m}$ is a a Gaussian distributed random variable. It is also well known that the normalized variable (N-1)$\hat{v}/\sigma^2$ is chi-squared distributed with N-1 degrees of freedom, or equivalently is Gamma distributed with scale parameters = 2 and shape parameter = (N-1)/2 (which has a mean = (N-1) and a variance = 2*(N-1)). If we let m_T and v_T be the mean and variance estimated from the target box, respectively, and m_B, v_B be the mean and variance from the background ring (generated using Eqs. (1) and (2)), then we can define the following metrics.

$$meanmet = \frac{|m_T - m_B|}{\sqrt{\frac{v_B}{N_T}}} \quad (5)$$

$$varmet = (N_T - 1)\frac{v_T}{v_B} \quad (6)$$

where N_T is the number of pixels in the target box. If m_B and v_B were the "true" background statistics, then if the target box had the same statistics as the background box meanmet (without the absolute values) should be a zero-mean, unit variance Gaussian random variable and varmet would be a Gamma-distributed random variable with mean = (N_T-1) and variance = 2*(N_T-1). However m_B and v_B are themselves random variables of course, so the actual distribution of meanmet and varmet is more complicated.

The other statistics that we want to use are the minimum and maximum values within the target box. This is driven by the fact that whale signatures often contain significantly brighter (or darker) values than the background. We can formally do this as follows. We are testing whether the target pixel values are Gaussian distributed with mean m_B and variance v_B. We can calculate the largest expected value within the target box by using the cumulative distribution function, F(T), defined as F(T) = Prob{X≤T} where Prob{} is the probability of what is inside the brackets. If we find the threshold, T_{MAX}, such that F(T_{MAX})=1-1/(2N_T) the expected number of samples within the target window that have values > T_{MAX} will be 0.5. In other words, we would not expect to see any samples greater than T_{MAX}. Similarly we can define a T_{MIN} such that F(T_{MIN})=1/(2N_T) and then we do not expect to see an samples less than T_{MIN}. Our metric will then be a count of the number of pixels within the target box that are either greater than T_{MAX} or less than T_{MIN}. To calculate these thresholds we actually need the inverse of the Gaussian cumulative density function, $F^{-1}(p)=T$; i.e. for a given p we need to find T such that F(T)=p. There is a good rational approximation to the inverse of the zero-mean, unit-variance Gaussian inverse CDF as

$$F^{-1}(p) = -5.5310\left\{\left(\frac{1-p}{p}\right)^{0.1193} - 1\right\} \; if \; p \geq 0.5 \quad (7)$$

$$F^{-1}(p) = +5.5310\left\{\left(\frac{p}{1-p}\right)^{0.1193} - 1\right\} \; if \; p < 0.5 \quad (8)$$

so that

$$T_{MAX} = \sqrt{v_B} F^{-1}\left(1 - \frac{1}{2N_T}\right) + m_B \quad (9)$$

$$T_{MIN} = \sqrt{v_B} F^{-1}\left(\frac{1}{2N_T}\right) + m_B \quad (10)$$

The metrics then become:

$$Maxmet = Number\ of\ pixels\ within\ the\ target\ box \geq T_{MAX}$$

$$Minmet = Number\ of\ pixels\ within\ the\ target\ box \leq T_{MIN}$$

So far we have discussed generating metrics for a single image. For the multi-spectral data we have multiple images, or bands, in each collection. There are a number of ways that we can deal with this. One would be to apply these metrics to each band, and then take the maximum metric across all the bands; i.e. declare that a target box is statistically different if a metric in any band passes the threshold. We could also average the metrics across bands. But usually it is the red, green, and blue bands that are visually used to find whale signatures, and when this is done the resulting images are not often colored – meaning that the red, green, and blue bands all have about the same value. So a more efficient approach is to form one image as the sum of the red, green, and blue bands for each multi-spectral image, and apply these metric to the summed image. This latter method has been used for the results in this paper.

We now need to define thresholds for the four metrics, meanmet, varmet, maxmet, and minmet, such that if they exceed the threshold then the target box is statistically different than the background ring. It is important to note that this anomaly filter is not meant to be a detector. Rather, it is meant to be a method to significantly downsize the number of chips that a user has to review in order to find the whale signatures within an image. Thus the thresholds need to be set such that the user does not have too many image chips to review; we generally found that we needed to limit the number of chips to around 8000 in order to allow a user to perform a review in 10-15 minutes. Thus we built the filter to have two passes. The first pass uses the image data to calculate a list of metric threshold and the resulting number of image chips that would pass the thresholds. The second pass is then done with the set of thresholds that generate around 8000 image chips. This will be useful only if the whale signatures make it through the second pass. To test that, we ran the anomaly filter on the Valdes 2012, Valdes 2014, and Witsand 2009 images over the regions where whale signatures were manually extracted. The anomaly filter found 226 of the 228 Pan signatures that had been manually extracted; so it only missed 2 signatures. It also found an additional 27 signatures within these images that visually appeared to be possible whale signatures but were not found with the manually analysis. It also found 130 of the 132 MSI signatures that were manually extracted, and found the same set of 27 new signatures. Thus it appeared that this approach was useful to allow a user to efficiently search for whale signatures in large images. Finally, we applied the filter to the Cape Cod 2021 Pleiades image and found an additional 54 MSI signatures.

The final set of whale signatures comprised of the manually extracted image chips plus the new ones found with the anomaly filter. We noted above that we had both Ortho and Basic WorldView images. The Basic images are rotated and scaled versions of the Ortho images, so to increase the test set size we also included image chips from both the Basic and Ortho images for the WorldView sensors. All of these together generated 730 whale chips across all the images. For reference, Appendix A shows the whale signature chips within the test set for both the Pan and MSI images; for the WorldView images only the Ortho results are shown.

We had no ground truth for any of this data; all the whale chips were extracted using visual analysis of the imagery. This means that a subset of the whale chips are probably not actually whales; they could be small boats or judiciously arranged breaking waves. The false alarms rates presented here are therefore potentially too low in that we are possibly including what should be a false alarms (small boat or breaking wave) as a whale signature detection by including it in the set of whale signatures.

Finally, we visually extracted background image chips from all of the images in regions that were far from where whale signatures were extracted and that covered the range of different types of background ocean responses that we could see within the images. This generated 9300 background chips across all of the images.

4. Feature Vector Values

As mentioned above, applying a feature vector based detection algorithm is only as good as the features being used; so we anticipate that we could improve detection performance as we developed better features. Because the anomaly filter values discussed above appear to have some skill in finding whale signatures we wanted to include them in the feature vector. This means that we need to preserve the nested boxes structure shown in Figure 1 for generating the features.

For this analysis we generated three sets of features.

(1) Normalized statistics within the target box. Specifically;

- m_T/s
- s_T/s
- $E[T^3]^{1/3}/s$
- $E[T^4]^{1/4}/s$
- $E[(T-m_T)^3]^{1/3}/s$
- $E[(T-m_T)^4]^{1/4}/s$

where m_T and s_T are the mean and standard deviation of the image values within the target box, E[] represents the expected value of what is inside the brackets, and T represents image pixels within the target box. The variable s represents the normalization factor. We looked at three possibilities: s=1 which means we are not normalizing the statistics and ignoring changes in scale factor; s= mean of the Coastal Band if it exists in the imagery, or the Blue band if there is no Coastal Band, or the Pan band if we are using panchromatic imagery within the target box; s=

mean of each individual band within the target box. We note that for the two options other than s=1, the first metric is just equal to 1 for the band being used for normalization so it is removed from the vector (i.e. if we are normalizing using one band, either Coastal or Blue, we remove it from that band's set of metric; if we are normalizing each band by its mean, then we remove it from all of the bands).

(2) The four anomaly statistics discussed in the previous section, slightly modified as follows;

- $(m_T-m_B)/s_B$
- s_T/s_B
- maxmet as defined above
- minmet as defined above

where m_B, s_B are the mean and standard deviation from the background ring.

Neither of these feature metric sets are tuned to the actual whale signature components. That is, they are both just general metrics that could apply to any target signature. So we also added 4 metrics that are more tuned to how whale signatures actually appear in the panchromatic imagery. Two of these metrics come from the observations that the whale signatures often contain regions of bright or dark areas that have some spatial extent, whereas a large majority of the background signatures do not contain such regions. One measure that captures this difference is the autocorrelation function of the image chip. We expect that the autocorrelation of whale signatures will be broader (i.e. have a wider peak) then the autocorrelation of the background due to these spatial regions of "similar" pixels. In addition, whale signatures tend to be elongated in one direction (i.e. whales are longer than they are wide) which translates to autocorrelation functions that are also elongated in one direction.

To implement these panchromatic metrics we first calculate the autocorrelation of the inner target box. We zero out the zero-lag value, since this is almost always significantly larger than all the other values. We then count the number of autocorrelation pixels that are ≥ 0.5 times the peak value in the function (after zeroing out the zero-lag value) and divide by the total number of pixels in the target box. This is a measure of what percentage of pixels in the image chip are correlated; i.e. a measure of the size of correlated image patches. To measure the elongation of the image, we generate a mask from the autocorrelation function of all pixels that are ≥ 0.5 times the peak value and fit an ellipse to the mask. We then calculate the ellipticity of that ellipse. This a metric for the elongation of a signature; the more elongated it is the higher the ellipticity of the autocorrelation mask.

Our third new metric is based on the observation that whale signatures often have bright/dark pixels right next to each other; particularly along the sides of the whale. To measure this we determine the maximum value of the absolute difference between successive lines or successive columns in the target box then divide this by the standard deviation of the pixels in the target box. This is a metric of whether the difference between successive pixels is dominating the variance of the image chip (which we expect it should if the major variation in the signal comes from these bright/dark pairs).

Our final, fourth, new metric is based on the fact that a whale signature will tend to have a small number of local maxima in the image chip, whereas one of the largest sources of false alarms, breaking waves, will often have multiple, small, bright blobs within the image chip. One method to measure this is to count the number of local maximum within the chip. For this analysis we define a local maxima as being the largest value within a 11x11 pixel window.

The four final feature vector entries as thus:

- corrcnt = number of pixels in the target box autocorrelation function that are ≥ 0.5*peak(without the zero-lag) divided by the total number of pixels in the target
- boxdelmax = maximum value of the absolute difference between successive lines or columns in the target box divided by σ_T
- ellip = ellipticity of the ellipse fit to the mask of the target box autocorrelation values that are ≥ 0.5*peak(without the zero-lag).
- loccnt = number of local maxima within the inner target box where local maxima are calculated over a 11x11 pixel window.

These last four features were derived from manual analysis of panchromatic whale signatures which had the resolution to see more detailed shapes of the signatures. The MSI signatures have approximately 4 times the panchromatic resolution and thus almost always appear more as bright blobs without any distinguishable shapes (see Appendix A). Thus for the MSI we can only rely on the spectral values and not the signature shapes, whereas with the panchromatic imagery we only have one band, but can rely on differentiating shape. Therefore we only used these last four feature elements when analyzing panchromatic imagery; for MSI we only used the first ten features described above.

The size of the feature vector varies depending on the normalization used and whether we are analyzing Pan or MSI imagery:

- if s=1, for Pan imagery the feature vector has 14 elements, for MSI it has 10 elements per band being analyzed;
- if s=mean of either Coastal, Blue, or Pan bands, then for Pan imagery the feature vector has 13 elements, and for MSI imagery it has 10 elements for each band except for the normalizing band which has 9 elements;
- if s=mean of each band, the for Pan imagery the feature vector has 13 elements (this is actually the same option for Pan as the previous one), and for MSI imagery it has 9 elements for each band being analyzed.

The final set of parameters to be determined is the sizes of the nested boxes in Figure 1. We want the target box to be the size of the whale signature, the guard box to have the same width to avoid contamination, and the background ring to be as large as possible but not to intersect with a neighboring whale signature. Based on manual analysis of the imagery, we have chosen the following box sizes:

- for MSI the target box is 8x8, the guard box is 16x16, and the background box is 32x32;

- For PAN imagery the target box is 32x32, the guard box is 64x64, and the background box is 128x128.

The training/testing set was then derived by centering these nested boxes over each of the manually derived whale signatures and generating the feature vector elements for each, and centering them over the manually extracted background signatures and generating those feature elements. This set of feature vectors for whale and background signatures became the training/testing set for the detection algorithms.

5. Optimal Linear Combination of Feature Vector Elements to Categorize the Feature Vector

The general problem we will address is as follows. We assume that we have an image that contains samples from N different classes, where each class represents some specific "thing" in the image such as forests, ocean, buildings, whales, etc. We want to classify each image pixel into the class that it belongs to using some feature vector to determine what class that pixel should be put into.

We will assume that for each class, the features that are generated from the class occupy a convex shape in M-dimensional space, where M is the number of features in the feature vector. In other words, an M-dimensional ellipse can efficiently enclose the features from a class; they are in essence a "blob" in M-dimensional space. As an illustration the top plot in Figure 2 shows schematic examples of three classes whose features occupy the spaces marked by the blue ellipses in the two-dimensional feature space. For visual simplicity these figures represent cases were we only have two features; i.e. the feature vectors are in two-dimensional space. All of the classes in the top plot obey our assumption of "blob-like". The region occupied by Class 4 in the bottom plot of Figure 2 does not obey our assumption; the horseshoe shape is not convex. Generally we do find that most real-world images have classes that generate blob-like feature clusters that exist within M-dimensional ellipses; however if a particular example does have more "horse-shoe" shaped classes than different approaches would have to be used.

Our assumption of blob-like clusters for each class is what allows us to efficiently perform the image classification using a linear approach. What we will do is find a set of hyper-planes that "optimally" separate each pair of classes, where we will define below what we mean by "optimal" in this context. Equivalently to finding hyper-planes is to find a set of what we will call classification vectors for each pair of classes where the vector is orthogonal to the hyper-plane. Let $\vec{c}_{kj}$ represent the classification vector between classes k and j. Each classification vector will have an associated threshold value, T_{kj}. For a given image pixel, we will dot-product the feature vector that is associated with that pixel with each classification vector; if the resulting scalar is less than the threshold we assign that pixel to the jth class, if it is above we assign it to the kth class. Since the classification vector is the vector that is orthogonal to the hyperplane that would separate the two classes, the projection of the feature vector onto this classification vector (i.e. the dot-product) is a measure of how far we are from the hyperplane on either side (see Figure 3).

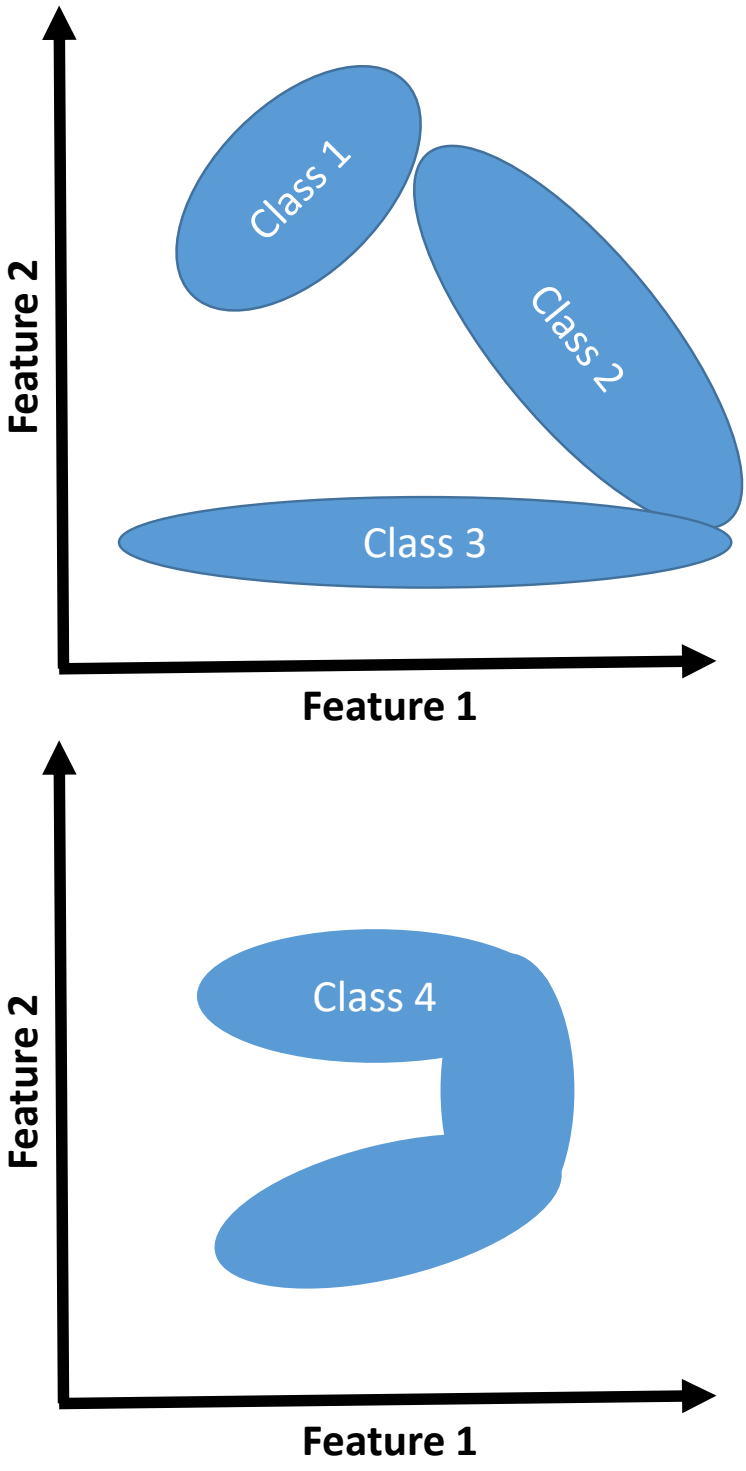


Figure 2: Schematic shown the types of feature distributions we are assuming (top) versus non-convex ones (bottom).

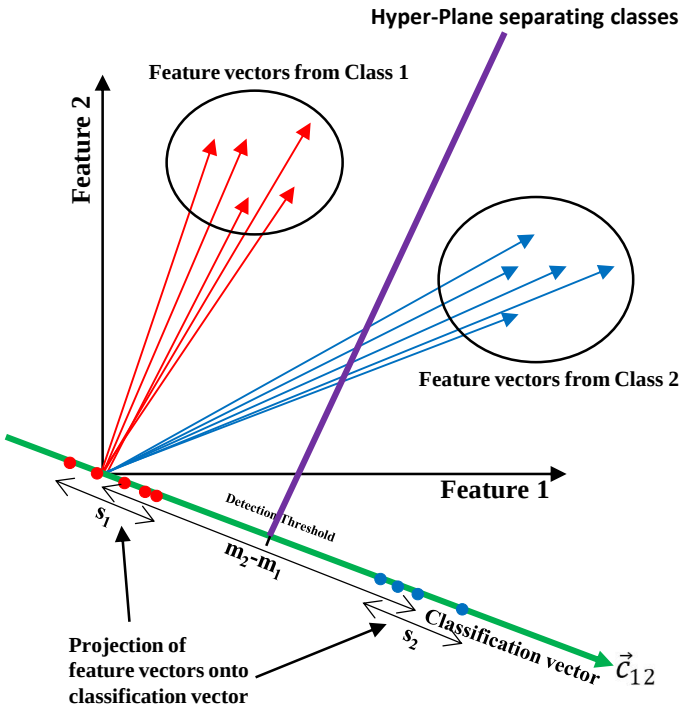


Figure 3: Schematic showing the relationship of the feature vectors from two classes to the classification vector and the hyper-plane separating the two classes. For simplicity the figure assume that feature vector only as two elements.

We want to define the classification vectors such that the scalars that result from the dot product of the classification vector with the features vectors from each class are a separated as they can be. We will generate these classification vectors using a training set (this is why this approach is a supervised segmentation) of example feature vectors derived from each class. Let s_k represent the scalar values generated from the dot product of the feature vectors from the training set for class k and the classification vector $\vec{c}_{kj}$, and likewise let s_j represent the dot product values between the class j feature vectors and $\vec{c}_{kj}$. We want to define $\vec{c}_{kj}$ such that we maximize the distance metric, d, defined as

$$d = \frac{\left(E[s_j] - E[s_k]\right)^2}{\left(var[s_k] + var[s_j]\right)} \tag{11}$$

where E[] and var[] represent the mean and variance, respectively of the values within the brackets. This distance metric is often referred to as the Fisher Discriminant (Lachenbruch, 1975) and one can see that as it increases the scalar values from the two classes get further apart.

There are other metrics that can be chosen to determine separation, but this choice has the advantage of the following analytical solution.

We can re-write Eq. (1) as

$$d = \frac{\vec{c}_{kj}^{T} M \vec{c}_{kj}}{\vec{c}_{kj}^{T} \left(C_k + C_j \right) \vec{c}_{kj}} \tag{12}$$

where M is a matrix defined as

$$M = \left(\vec{m}_j - \vec{m}_k \right) \left(\vec{m}_j - \vec{m}_k \right)^T , \tag{13}$$

and where $\overrightarrow{m_k}$ is the mean feature vector from class k, $\overrightarrow{m_j}$ is the mean feature vector from class j, C_j is the covariance matrix for the feature vectors from class j and C_k is the covariance matrix for the feature vectors from class k. We can re-write Eq. (12) as an eigenvalue problem

$$\left(C_k + C_j \right)^{-1} M \vec{c}_{kj} = d \vec{c}_{kj} \tag{14}$$

where the -1 superscript indicates matrix inversion. The classification vector we want is the eigenvector that generates the maximal eigenvalue and thus maximizes the distance metric d. Since the matrix M has unit rank, Eq. (14) has only a single eigenvector solution which can be written as

$$\vec{c}_{kj} = \left(C_j + C_k \right)^{-1} \left(\vec{m}_j - \vec{m}_k \right) \tag{15}$$

which generates a distance metric value of

$$d = \left(\vec{m}_j - \vec{m}_k \right)^T \left(C_j + C_k \right)^{-1} \left(\vec{m}_j - \vec{m}_k \right). \tag{16}$$

which is the classic linear discriminant result (Lachenbruch, 1975). The optimality mentioned above is that this approach generates classification vectors (and thus hyperplanes) that maximize the distance metric d in Eq. (11).

If the features in our feature vector are uncorrelated, then the covariance matrices C_j and C_k only have diagonal elements, so the inverse of the sum in Eq. (15) is just the inverse of the sum of the variances along the diagonal. Thus in this case the optimal classification vector is just

$$\overrightarrow{c_{kj}} = \left[\frac{\overrightarrow{m_j}(i) - \overrightarrow{m_k}(i)}{\left(\sigma_k^2(i) + \sigma_j^2(i) \right)} \right] \text{ for i from 1 to M} \tag{17}$$

where $\sigma_k^2(i)$ is the variance of the i-th feature from class k and $\sigma_j^2(i)$ is the variance of the i-th feature from class j.

In the general case, we apply each classification vector to a given feature vector and record which class it gets put into for each class pair. We assign that pixel to a given class only if it gets put into that class for each pair of classes that contain that class. For example, if there are four classes, and a given feature vector gets put into class 1 when using classification vectors for class pairs 1-2, 1-3, and 1-4, then it is put into class 1. Note that if this is true for one class, it can not be true for any other class. It may be that this is not true for any class, in which case we can either not classify that pixel (i.e. give it a "No Class" designation), or assign probability to which class it is in instead of making the classification binary. That is, count how many times it got assigned to each class and use this to assign a probability that it belongs to a given class.

We had mentioned above that to determine which class a feature vector belonged to, we compared the dot-product with a classification vector to a threshold. In application, it is also useful to define an interval within which the dot-product scalar must lie instead of just a threshold; i.e. to be in class k it must be greater than T_{kj} and less than some maximum value. This eliminates the problem that when we compare to two classes that are far away from the actual class the feature vector belongs into we are getting scalar values that are very far away from the hyperplane and way outside of where that scalar should lie for a given class.

This general approach has been successfully used to do automated ice-type classification in space-based Synthetic Aperture Radar (SAR) imagery (Wackerman and Miller, 1996; Wackerman et al., 2004) as well as terrain classification in airborne SAR imagery (Wackerman 1997)

If there is only one class that we really care about finding, then this approach collapses to a detection problem where there is a target class and a background class. However, we note that even in this case, it can be useful to have multiple background classes even if we only care about detecting a target. This is illustrated in Figure 4 where we show that there is clean separation between the target class and each of 3 background classes, but if we were to lump the background classes into one class, we would not have clean separation (and the background class would violate our ellipsoidal assumption). So some care must be given in understanding the nature of the background we are trying to find a target in.

For the detection case where we have one background class we can generate a Receiver Operating Characteristic Curve (ROCC) which plots the probability of detection versus probability of a false alarm as follows. From the training set of target feature vectors and background feature vectors we derive the optimal classification vector as described above. We then dot-product the classification vector with each feature vector in a separate testing set to generate a series of scalars from the target set and one from the background set. We can then form a density function from these values for each class by forming a histogram and normalizing it to sum to 1. For each histogram bin we sum the background histogram to the right of that bin to generate a probability of a false alarm (PFA) and we sum the target histogram to the right of that bin to get a probability of detection (PD). By doing this for all the histogram bins from smallest to largest we map out the ROCC. This allows us to compare ROCCs for different choices of features; the more that ROCC moves to the upper, left the better those choice of features are. Note that the detection metric d in Eq. (12) itself is also a measure of how good the choice of features are; the larger this number the better the detection of the target. However, it is

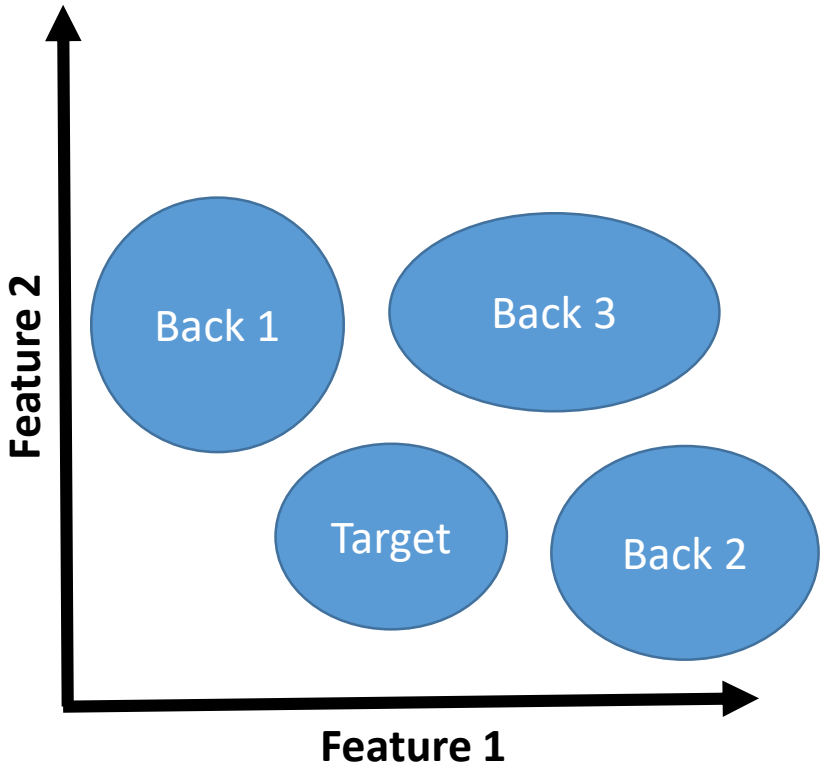


Figure 4: Samples of needed to separate background responses into different classes to provide better separation from a target class.

important to keep in mind that the metric d does not itself completely determine the shape of the ROCC; so the ROCC is still the better comparison.

To generate more robust performance estimates we do N trials with the test data, where for each trial we randomly divide the data into half training and half testing, generate the classification vectors using the training data, then generate the ROCCs using the testing data. We then record the ROCCs for each trial, as well as the ROCC if we use all the data to both test and train – which generates the best performance possible. In the results below we will use 10 trials for each set of data, and compare the resulting 11 ROCCs (10 from each trial plus the one from using the full set of data) for consistency of performance. The region occupied by all these ROCCs then indicates the performance region for the algorithm.

6. Using a Neural Net to Categorize the Feature Vector

Another approach to determining if a feature vector comes from a target or background region is to use a neural net. A neural net is a classic AI or deep learning algorithm that attempts to model what the neurons do in your brain. A neural net consists of layers of "neurons" which have different weights. The feature vector is put into the first layer of the net, then the data is moved through the net until it generates output weights that determine which class the vector belongs to.

Training the net involves determining what the weights need to be to minimize the error of the output.

There are many parameters that have to be chosen to design a given neural net. If we treat this as a binary classification problem (the input data is being put into one of two classes) this means that the last layer of the net must have two neurons (representing the "probability" or weight of it being in one or the other class). If we treat it as a regression problem, where we assign a value of 0 to all background signatures and a value of 1.0 to all target signatures and then try to develop a neural net that estimates this value, then the last layer needs to have one neuron which gives us the estimate of the signature "value" (i.e. with 0 for background or 1 for target). Also, the first layer must have the same number of neurons as elements in the feature vector. The other parameters we get to choose are:

- How many other layers (often called "hidden layers") are in the net. We looked at 1 to 3 hidden layers with the results being essentially the same, with only slightly better PD results for 3 layers, so that is what is used for these results.

- How many neurons are in each of the hidden layers (remember that the number of neurons in the first and last layer are subscribed by the problem). A common approach is to make the number of neurons either the same as the input, or twice. We looked at both options and found no real change in performance, so we used twice the number of neurons in the first layer (i.e. the number of elements in the feature vector) for each hidden layer.

- How to measure the "error". This is often coded as an estimate of information loss. For binary classification, the most common lost function is referred to as Cross Entropy which calculates the loss as the sum of x*log(y) where x is the "label" for the signatures (i.e. either 0 or 1) and y is the output estimate that has the larger weight (and thus determines which class that signature is put into). If we treat this as a regression problem, then the most common error function is mean-squared-error (MSE) between the signatures label (again either 0 or 1) and the estimate of that label. These are the two we used.

- What algorithm to use to estimate the weights. This is some version of a search algorithm that iterative changes the weights so as to minimize the loss function. One of the more common ones is Stochastic Gradient Descent, which we used.

- There is a "learning rate" that is a scalar that determines the speed at which the weights are "learned". A very common value is 0.001 which we used. We note that this learning rate was used in (Guirado et al., 2019) and it generated the second best result in (Borowicz et al., 2019).

- To determine weights, the algorithms divide the training set into sub-sets of training and testing data. The user specifies the size of these subsets, often referred to as the batch size. We looks at batch sizes of 64, 100, 200 and found that it did not impact the final performance at all, so we used a batch size of 64.

- How many iterations to perform to estimate the weights. This is often referred to as the number of epochs. Typically we want to set this to a number such that the decrease in information loss has plateaued, and/or the number of correctly classified signatures has plateaued. We found that for the binary classification problem (using the Cross Entropy information loss function) this was around 300 iterations. For the regression problem (using MSE as the information loss function) that had to be around 500 iterations.

- A function to determine if the neuron passes data on or not. This is essentially modeling whether the neuron "fires" to pass data onto the next level, and are referred to as activation functions. If we do not have this, then the net because essentially a linear model; each neuron scales its input data by some weight and passes it on. Inserting a "firing" function introduces non-linearity into the network. Two common type of activation functions are: (1) only pass positive values, known as the rectified linear unit function; or (2) a function that normalizes the output to a value between 0 and 1 (to do this there is a sigmoid function, an exponential weight function (called softmax), or hyperbolic tangent). We chose the rectified linear unit function, however we also generated a net without any firing function, and the performance was essentially the same.

Having defined the network, we then trained it by randomly dividing the target and background feature vectors in the training set into a training set and a testing set. The algorithm then divides the training/testing sets into batches of the size determine by the user (in our case 64) and does an iterative estimation of the weights by using the training set to determine the next set of weights and the testing set to make sure that the iteration is improving things. Since we randomly divide the original training data into training and testing sets, each time we run the code we actually generate a new neural net and a new set of performance values. If we are using the binary classification problem, then the final neural net classifies each training set signature into either a target or a background signature, so we get a single set of PFA, PD values for a given net. If we are using the regression problem, then the final neural net generates an estimate of the signature "value" for each signature. We can then assume some threshold for this estimate and label as a target every signature whose value estimate is above this threshold and label as a background every signature that is below this threshold. Thus by changing the threshold we generate a series of (PFA, PD) values, similar to how we generate the ROCC with the optimal linear algorithm. For this analysis we used 10 trials and thus generated 10 ROCCs, similar to what is generated using the optimal linear algorithm.

This is a significantly simpler neural net than what is normally used in standard CNNs, and what was used in both (Guirado et al., 2019) and (Borowicz et al., 2019) when they trained on hundreds of thousands of images from image databases on the internet. However in this study we wanted to see if a simple neural net applied to feature vectors could at least start to get close to the performance of the much large CNNs trained on much larger training sets.

7. Performance on Test Set

Our first analysis is to determine detection performance across the entire test set. For MSI the sets of bands that all the images share are the four bands B, G, R, NIR, so we just used these bands in each MSI. We looked at the three different normalization approaches (which collapse to two for PAN) and looked at results for the optimal linear combination, the neural net considered as a regression problem, and the neural net considered as a binary classification problem.

In addition, we looked at using the image values within the target box as the feature vector, instead of the features described above. The reason for this is that using the image values in the neural net is an approximation for what a CNN would perform and gives us a benchmark to compare the feature vector results to. It also tests to see if we are improving performance for these small test sets by using feature vectors instead of imagery values. Due to the simplicity of our neural net, our results will not be comparable to what a significantly more complicated CNN would be able to provide, but rather it is providing a comparison between using feature vectors and using image value for the same test data and simple neural net.

Figure 4 shows these results. The first line of plots in Figure 4 are for the MSI using only the consistent 4 bands and using no normalization (left plot), normalizing by the Blue band (middle plot), and normalizing by each bands mean (right, plot). The black line are the 10 ROCCs from the optimal linear algorithm. The green-dashed line is the ROCC from the optimal linear algorithm using the full data set. The red lines are the 10 ROCCs from the neural net algorithm using regression. The blue dots are the 10 (PFA,PD) results from the neural net algorithm using binary classification. The black number in the lower, right of each plot is the average number of false alarms per WorldView Tile (NFA/Tile) for a PD=90% for the optimal linear algorithm, where the averaging is over the 10 trials. The red number is the average NFA/Tile at PD=90% for the neural net regression algorithm, again averaged over the 10 trials. Note that the PD values along the y-axis cover the full spread of possible data (0 to 1.0) whereas the PFA values along the x-axis are only the values near to the origin (0.0 to 0.04) to allow the ROCCs to show where they have the bend in the curve. The second line of plots are the same results but for the full set of panchromatic images with either no normalization (left plot) or normalized by the mean of the image (middle plot). Finally, the last two lines of results are for the same image sets as the top two lines, but for the feature vector being the image values within the target box instead of the features. For all of these results the neural net binary classification algorithm did not converge. For the left plots (no normalization) the neural net regression algorithms did not converge. For the other plots, we only show 3 trials for the neural net regression algorithm due to the large run-time required to generate these results.

There are a number of general comments we can derive from the results in Figure 4.

- The neural net binary classification using the network parameters we list above did not ever work. When it converged it only gave PDs of around 50%, and it never converged when using image data as the feature vector. In future work we will see if changing the neural net parameters can generate better results.

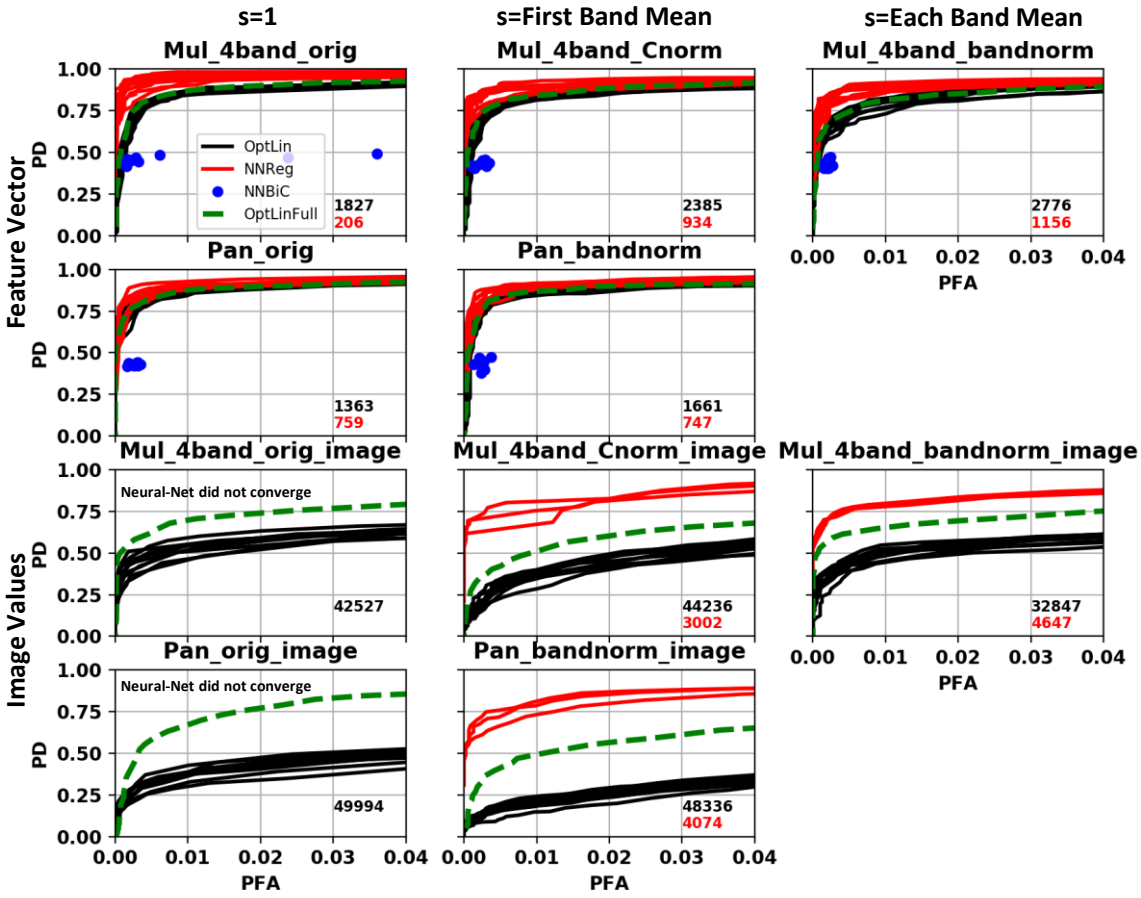


Figure 4: ROCC and NFA/Tile results for various image combinations and algorithms. Red lines are ROCCs from the neural net regression algorithm trials. Black lines are ROCCs from the optimal linear algorithm trials. Dashed green line is the ROCC using the optimal linear algorithm on the full test set. Blue dots are the PD, PFA results for the neural net binary classification algorithm. The columns are results for different normalization approaches; left column does not perform any normalization; middle column divides the imagery by either the Coastal, Blue, or panchromatic band depending on the image; right column divides each band by its mean. The first line of plots are for MSI images from all the data using only the common four bands of B,G,R,NIR. The second line of plots is for all of the Pan images. These results all use the feature vector described in the text. For comparison, the last two lines of the figure show results when the image values within the target box are used as the "feature vector". The neural net binary classification algorithm did not converge for any of these, so it is not plotted. The neural net regression algorithm did not converge for the left plots, so they are not plotted. Finally, the red numbers within each plot are the NFA/Tile values averaged over the trials for a PD=90% for the neural net regression algorithm. The black numbers are the NFA/Tile averages for the optimal linear algorithm.

- Using image data as the feature vector is consistently and significantly worse than using the actual features. This is most probably due to the fact that we have such a small training set that is not sufficient for standard CNN models. It does indicate that using the feature vectors can provide improved performance for these small test sets and simple neural nets.
- The neural net regression approach consistently outperformed the optimal linear algorithm. Their ROCCs are consistently above and to the left of the optimal linear ROCCs.
- For the feature vector results the cluster of ROCCs for the 10 trials is compact, and the full-data optimal linear ROCC is within the 10 trials, showing that for this data set the statistics we are generating are consistent across the test set. For the image values results, the trial data is still compact, but the full optimal linear results is significantly above the trial results, indicating that there is more variation in these statistics than when using the feature vector.
- The best performance is for the MSI 4-band imagery without normalization. It generates a NFA/Tile of 206; well within what would be reasonable for a user to review.
- The performances of the MSI 4-band imagery normalizing by the blue band, and both the normalizations of the Pan image, were essentially the same. The neural net algorithms generated a NFA/Tile of ~800.

We then compared using the full 8 bands of the WorldView imagery with either Ortho or Basic imagery. If we limit the test dataset to only WorldView imagery we then have only 340 whale chips. These results are shown in Figure 5. The top row of plots are for using all 8 bands of the Ortho imagery in the three different normalizations. The second row uses the 8-band Basic imagery. For comparison, the last row shows a result where we only used the 4 B, R, G, NIR bands of the WorldView Ortho imagery without normalization.

Some comments on these results:

- The WorldView Ortho imagery have the best performance across all normalizations, with no normalization again generating the best result; NFA/Tile = 12.
- The WorldView Basic imagery performs worse than the Ortho. This seems in part due to the DRA that is applied to the Ortho image to enhance the contrast of the image. This appears to improve the detectability of the whale signatures over the Basic image.
- The cluster of ROCCs is tight for the Ortho imagery, but is quite spread for the Basic, and the optimal linear full results for the Basic imagery is quite above the trial results. All this indicates that the statistics for the Basic imagery are more variably across the data set than for the Ortho imagery, again perhaps due to the DRA.
- Using only four bands in the WorldView Ortho images performs almost as well as using all 8 bands, indicating that it is not the additional bands that are giving us the significant performance boost but the nature of the Ortho images; again most probably due to DRA which is only applied to the Ortho images.

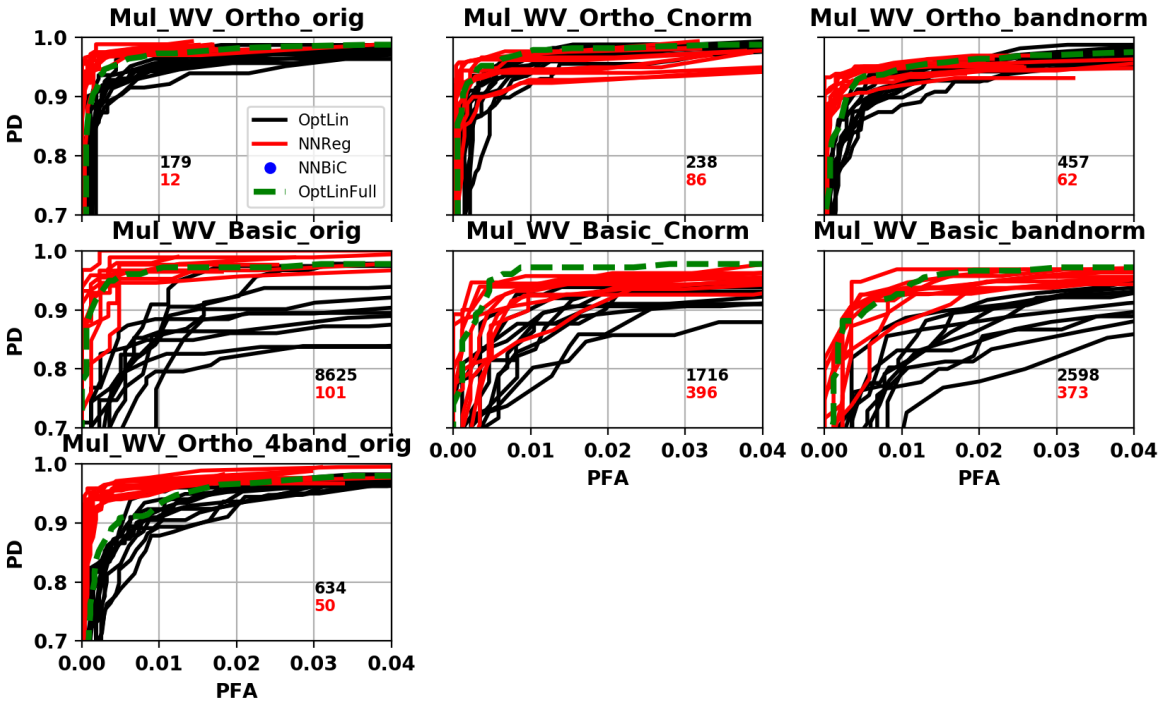


Figure 5: ROCC and NFA/Tile results for various image combinations and algorithms. Formats of the plots are the same as in Figure 4. In this figure, the top line of plots only use the WorldView MSI Ortho images and the middle line only use the WorldView MSI Basic images. In both cases all 8 bands are used. The bottom plot uses the WorldView MSI Ortho images, but only the four B,G,R,NIR bands.

Figure 6 adds the Pan results for the Ortho or Basic WorldView images (rows 2 and 4; rows 1 and 3 are the same results as are shown in Figure 5). For both Ortho and Basic imagery the MSI out-performs the Pan, even though we added the image features onto the Pan feature vector. Also, band-normalizing improved Pan performance over no normalization. Note that when we used all the images (Figure 4) the two Pan normalizations generated essentially the same result. Finally, Pan Ortho images outperform Pan Basic images, and both out-perform Pan applied to all the imagery.

To provide a summary for the different algorithms and imagery selections, we calculated the average ROCC over all trials for the variation combinations, and the average value for NFA/Tile over all trials. These are shown in Figure 7 for the various combinations. The notations around the plot indicate which image types were used and show the average NFA/Tile as a red number (this are the same numbers as are shown in Figures 4-6). The MSI results in Figure 7 use no normalization and the Pan results use normalization by the band mean.

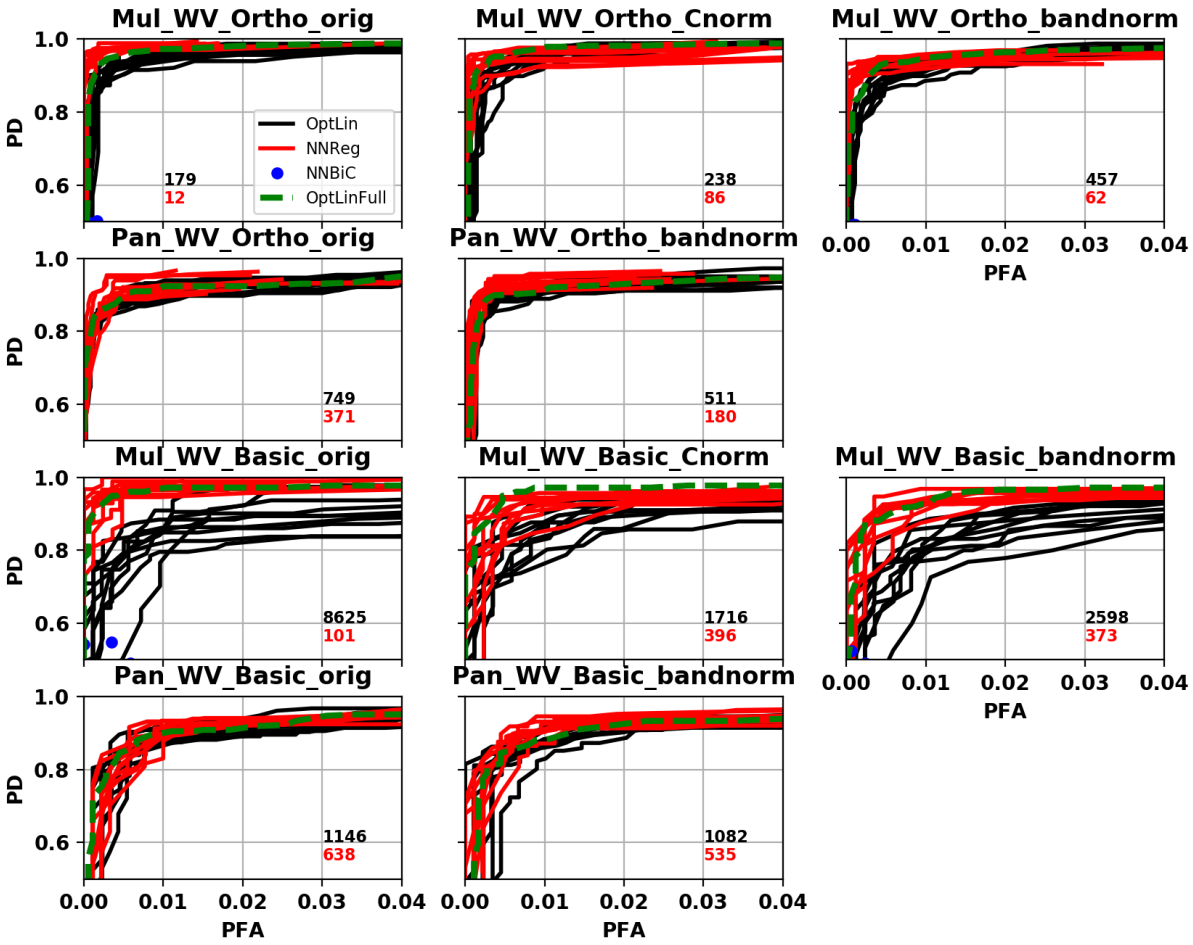


Figure 6: ROCC and NFA/Tile results for various image combinations and algorithms. Formats of the plots are the same as in Figure 4. In this figure, the top line of plots only use the WorldView MSI Ortho images and are the same results as the top line of Figure 5. The second lines uses the WorldView Ortho Pan images. The third line uses the WorldView MSI Basic images and shows the same results as in Figure 5. The bottom line uses the WorldView Pan Basic imagery.

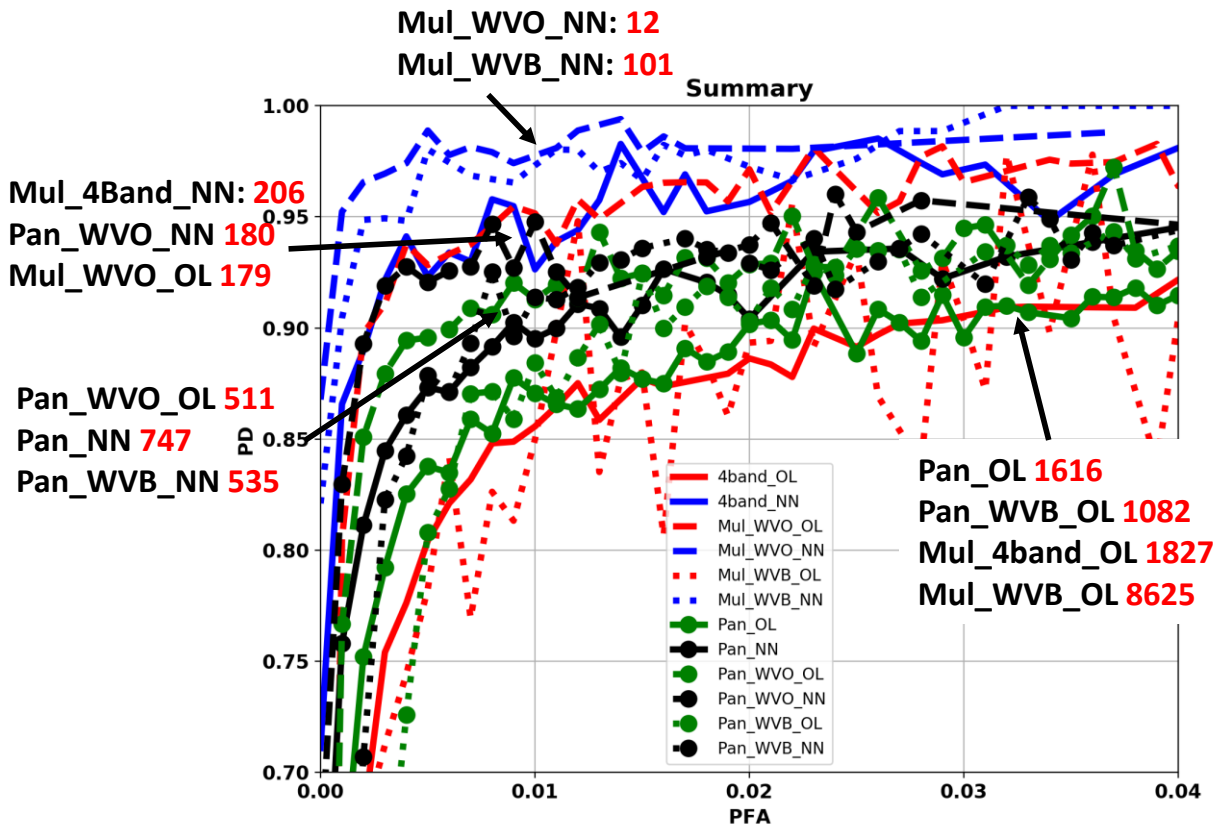


Figure 7: Summary of the ROCCs for the various combinations. For all the MSI results in this plot, no normalization was used. For all the Pan results, they were normalized by the mean within the target box. The notations around the plot show what images/algorithms each ROCC corresponds to (as does the legend) along with a red number that is the average NFA/Tile for a PD=90%.

The results fall into four main groups in Figure 7:

- Using WorldView MSI Ortho or Basic with the neural net algorithm are the best, top, results with NFA/Tile=12/101 and a PD~97% for PFA of 1% (0.01). Again, excellent performance and they stand out from the rest of the image combinations.
- The next group consists of (a) all MSI images using 4 bands with the neural net algorithm; (b) WorldView Ortho PAN imagery with the neural net algorithm; and (c) Worldview Ortho MSI with the optimal linear algorithms. These all generate NFA/Tile ~ 190 (from 179 to 206) with a PD~94% for PFA=1%. These false alarms could be easily reviewed by a user in an operational setting so we are still within operational utility.
- The next group consists of (a) WorldView Ortho PAN imagery with the optimal linear algorithms; (b) WorldView Basic PAN with neural net; and (c) All Pan imagery with the neural net algorithm. NFA/Tile ~ 600 (from 511 to 747) with a PD~90% for PFA=1%. Note that for this group the NFA/Tile is starting to get too high for a user to easily review in an operational setting.

- The final group consists of all optimal linear algorithms using (a) all Pan imagery; (b) WorldView PAN Basic; (c) all MSI imagery; and (d) WorldView Basic MSI. The NFA/Tile ~ 1400 (from 1082 to 1827 with the excepthon of WorldView Basic MSI that fluctuates wildly) and with a PD~86% for PFA=1%. Note that for this group the NFA/Tile is more than we would want a user to have to review in an operational setting.

Clearly we want to stay within the first two groups, which means that we want to use either :
(a) only WorldView MSI imagery (Ortho or Basic) or PAN Ortho with the neural net algorithm; or
(b) use any available satellite sensor system, in which case we want to use MSI with the neural net algorithm.

The results in Figure 7 also support the general conclusions of: (a) the neural net algorithm is better than the optimal linear algorithms (4/5 results in the top 2 groups are neural net and 5/7 of the bottom two groups are optimal linear); (b) MSI is better than PAN even with the additional PAN features (4/5 results in the top 2 groups are MSI and 5/7 of the bottom two groups are PAN)

Running either the optimal linear or neural net algorithms on the test set generates a detection algorithm that can be applied to a full image. In the case of the optimal linear approach, this consists of the classification vector and associated thresholds. In the case of the neural net approach, this consists of the weights that were generated along with the structure of the neural net. Note that each trial generates a different detection algorithm, so we pick the one that generated the best ROCC between the ten trials. To test the robustness of the test set results against the full imagery, we ran the detectors on the full WorldView Ortho Pan images for the Valdes 2012, Valdes 2014, and Valdes 2016 and on the full Geo-Eye Pan image for Witsand 2009. We used normalization by the mean of each band since this performed the best over the Pan test sets, and ran both the neural net and optimal linear detectors. We assumed that the manually-derived plus anomaly-filter whale signatures were the only whale signatures in the entire image and treated them as ground truth. So PD was determined by the number of these that were detected whereas PFA was determined by detection of anything else. The image results are shown in Figure 8 as diamonds with dashed lines and for comparison are plotted on-top of the ROCCs from Figure 7, though they are now plotted with thinner lines to make the image results stand out. All but one of the images generated PD ~93% for PFA in the range from 0.005 to 0.02 which spans the regions of group 2 and group 3 as discussed above. Note that these groups span the test set results for WorldView Ortho Pan images, either with neural net or optimal linear algorithms, and all Pan images with neural net algorithm, so the image results appear consistent with the test set results. The remaining image generated PD~ 90% for this same range using the optimal linear algorithm and is consistent with the lower portion of group 3; so again consistent but at the lower end. Overall, the full image results appear consistent with what the test set generated.

We also performed timing tests of the detection algorithms to see how quickly they could be used to generate results on imagery. Run time per WorldView tile ran from 500 to 1000 seconds, with the optimal linear and neural net detectors taking approximately the same time. A

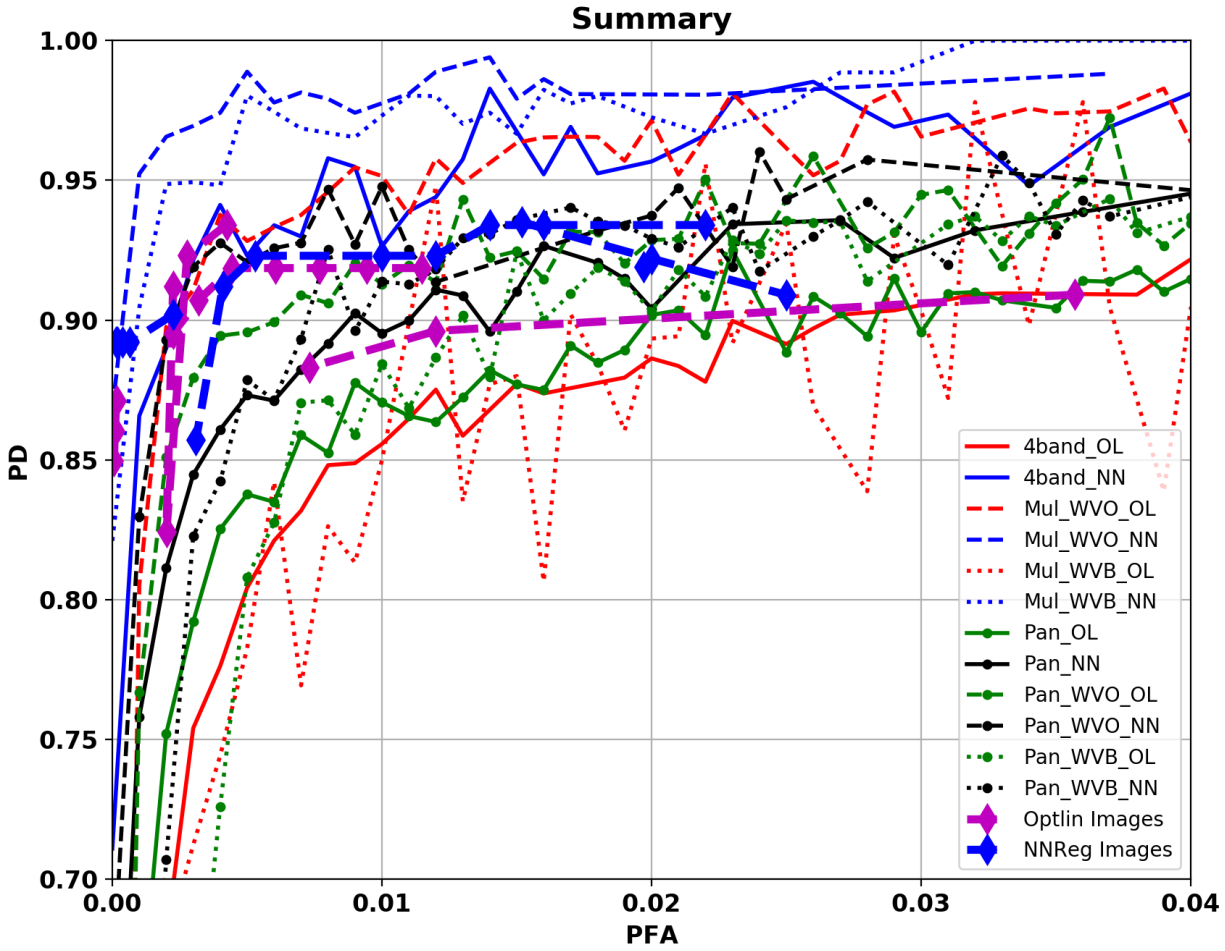


Figure 8: Same results as in Figure 7 (but drawn with thinner lines and smaller symbols) with the results from running the detection algorithm on full imagery added. Purple diamonds are full image results using the optimal linear algorithm. Blue diamonds are full image results using the neural net regression algorithm. The image results come from four images: Valdes 2012, Valdes 2014, Valdes 2016, and Witsand 2009 and were run over the tiles which contained the manually-derived whale signatures.

full WorldView collection can contain on the order of 30 tiles, so to process the full collection would take from 4 to 8 hours, which is too long for truly operational considerations. However the code implementing the algorithms has not yet been optimized for run time, so it may be possible to make it faster.

Finally, we compare our ROCCs with the results previously published. Figure 9 shows the ROCCs from Figure 7 with the results from (Fretwell et al., 2014; Borowicz et al., 2019; Guirado et al., 2019) shown as large purple circles. Note that we have expanded the PFA values along the x-axis in Figure 9 to go from 0 to 0.3 in order to show all the published results. With our testing set, the specific feature vector described in this report, and the specific simple neural net that we used, we do attain better performance than (Fretwell et al., 2014) and (Guirado et al.,

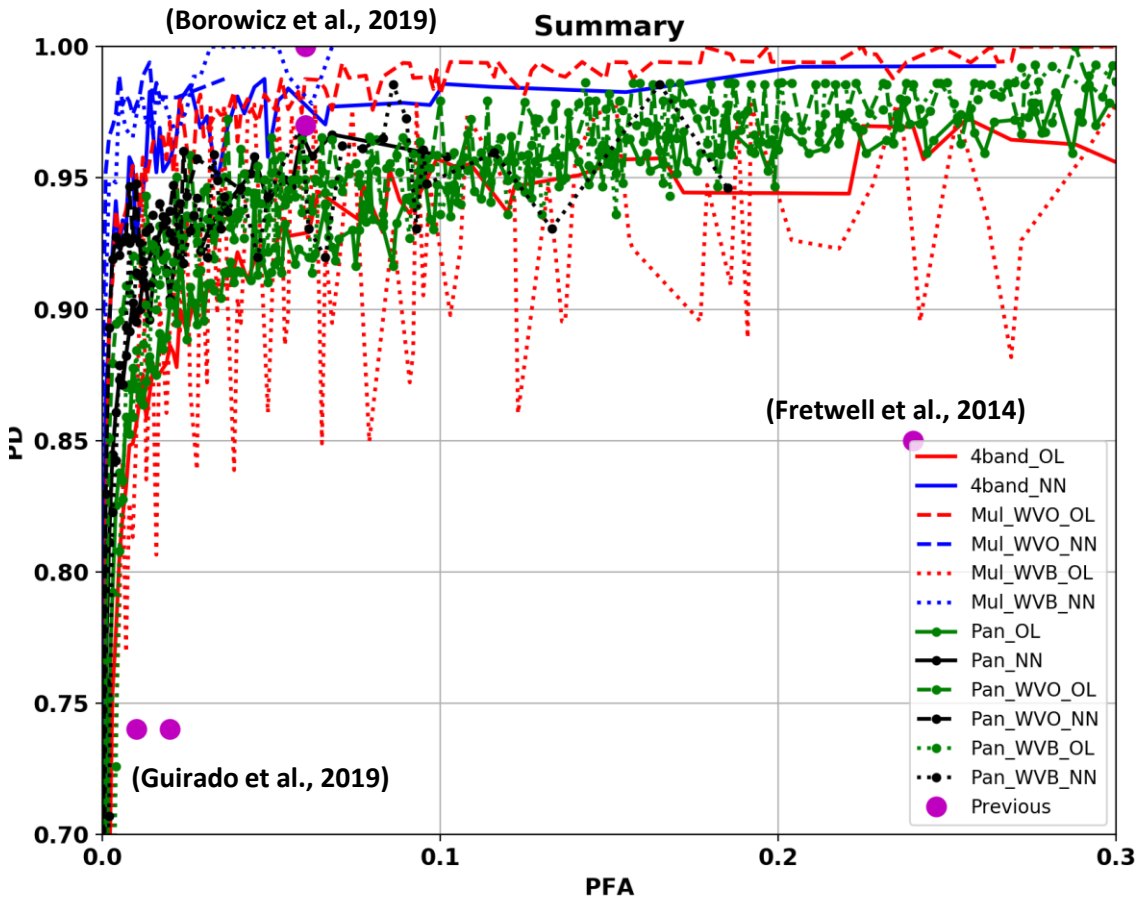


Figure 9: Comparison of ROCCs from Figure 7 to published results from three papers (large purple dots).

2019), but we appear similar to (Borowicz et al., 2019) in that their results straddle what we referred to above as Group 2 in our results, but they do achieve higher PD (100%). This is interesting because (Borowicz et al., 2019) used very large training sets from image databases and a much more complex neural net than we have used in this study, applying image values to a CNN. These results indicate that by using feature vectors with a much simpler neural net we can achieve detection performance similar to what the larger CNNs provide when being applied to image values.

8. Summary and Conclusions

In this paper we have presented an automated detection algorithm for whale signatures in optical satellite remote sensing imagery that is an attempt to design an approach that will perform better on smaller test data sets than the traditional deep learning CNN approaches. The detection algorithm combines a feature vector with either an optimal linear combination of feature elements or a neural net to decide whether the feature vector comes from a whale signature or not. The feature vector used in this study is a combination of normalized statistics, statistics designed to determine if the image values within some target box are statistically different than those in the local background, and a set of metrics designed to measure image properties unique

to whale signatures in panchromatic imagery. The proposed approach was tested on a set of image data that contain WorldView-2, WorldView-3, GeoEye-1, Quickbird-2, and Pleiades imagery from which 730 whale signatures and 9300 background signatures were manually extracted to provide training/testing data. Conclusions from this study are:

- Using a feature vector approach performed better than using image values directly into either the optimal linear combination or neural net algorithms, giving some indication that for small training sets a feature-based approach may provide improved performance over using image values;

- Our results were either better or equivalent to published results that used significantly larger training sets and significantly more complex CNN neural nets, again giving some indication that for small training sets a feature-based approach can provide improved or equivalent performance over traditional CNN approaches.

- Using a neural net to determine which class a feature vector belongs to performed better than using an optimal linear combination, and both ran in about the same time on an image;

- Using MSI gave better performance than using PAN imagery, even with the addition image metric used with PAN imagery;

- Using WorldView 8-band multi-spectral imagery the proposed algorithm generates a probability of detection of ~ 97% at a probability of false alarm of ~1% with 10-100 false alarms within a WorldView tile. Using 4-band multi-spectral imagery from all sensors, the proposed algorithm generates a probability of detection of ~ 94% with a probability of false alarm of ~ 1% with ~200 false alarms per WorldView tile.

- WorldView Ortho imagery provides the overall best performance, most probably due to the DRA that is applied to the image to enhance contrast.

As mentioned above, the training set was derived from manual analysis without any i*n situ* ground truth, so there is probably a subset of our "whale" image chips that are in fact not whales but small boats or judiciously placed breaking waves. This means that our reported PFA values are probably too low since we have put actual false alarms into the "whales" category. However without any ground truth it is difficult to determine how far off our PFA values may be.

The results shown here are specific to our test set, the set of features we used in the feature vector, and the specific parameters for the neural net. In future work we will determine if a more complex neural net will significantly improve performance, and we will continue to search for better features to use for whale signature detection. We also will apply the detection algorithms to full MSI Ortho images to determine if we can get image results consistent with the test set ROCCs.

References:

Abileah, R., "Use of high resolution space imagery to monitor and abundance, distribution, and migration patterns of marine mammal populations", Proc. Of, MTS 0-933957-28-9, 2001

Abileah, R., "Marine mammal census using space satellite imagery," U.S. Navy Journal Underwater Acoustics, 52, No. 3, 709-724, July 2002

Barber-Meyer, S.M., G.L. Kooyman, P.J. Ponganis, "Estimating the relative abundance of emperor penguins at inaccessible colonies using satellite imagery," Polar Biology, 30, 1565-1570, 2007.

Borowicz A., H. Le, G. Humphries, G. Nehis, C. Hoschle, V. Kosarev, H.J. Lynch, "Aerial-trained deep learning networks for surveying cetaceans from satellite imagery," PLoS ONE, 14(10), https://doi.org/10.1371/journal.pone.0212532, 2019.

Clarke, P.J., H.C. Cubaynes, K.A. Stockin, C. Olavarria, A. de Vos, P.T. Fretwell, J.A. Jackson, "Cetacean strandings from space: challenges and opportunities of very high resolution satellites for the remote monitoring of cetacean mass strandings, " Frontiers in Marine Science, 8, doi:10.3389/fmars.2021.650735, 2021.

Cubaynes, H.C., P.T. Fretwell, C. Bamford, L. Gerrish, J.A. Jackson, " Whales from space: four mysticete species described using new VHR satellite imagery," Marine Mammal Science, 35(2), 466-491, 2019.

Cubaynes, H.C., W.G. Rees, J.A. Jackson, M. Moore, T.L. Sformo, W.A. McLellan, M.E. Niemeyer, J.C. George, J. van der Hoop, J. Forcada, P., Trathan, P. Fretwell," Spectral reflectance of whale skin above the sea surface: a proposed measurement protocol," Remote Sensing Ecology and Conservation, doi: 10.1002/rse2.155, 28 Jan., 2020

Fretwell, P., P.N. Trathan, "Penguins from space: faecal stains reveal the location of emperor penguin colonies," Global Ecol. Biogeogr., 18, 543-552, 2009.

Fretwell, P.T., M.A. LaRue, P. Morin, G.L. Kooyman, B. Wienecke, N. Ratcliffe, A.J. Fox, A.H. Fleming, C. Porter, P.N. Trathan, "An emperor penguin population estimate the firsts global, synoptic survey of a species from space," PloS ONE, 7(4), doi:10.1371/journal.pone.0033751, 2012.

Fretwell, P.T., J.A. Jackson, M.J. Ulloa Encina, V. Haussermann, M.J. Perez Alvarez,C. Olavarria, "Using remote sensing to detect whale strandings in remote areas: the case of sei whales mass mortality in Chilean Patagonia," PLoS ONE doi: 10.1371/journal.pone.0222498, 2019.

Fretwell, P.T., I.J. Staniland, J.Forcada, "Whales from space: counting southern right whales by satellite," PLoS ONE, 9(2), doel10.1371/journal.pone.0088655, 2014.

Guirado, E., S. Tabik, M.L. Rivas, D. Alcaraz-Segura, F. Herrera, "Whale counting in satellite and aerial images with deep learning." Scientific Reports Nature Research, 9:14259,| https://doi.org/10.1038/s41598-019-50795-9, 2019.

Lachenbruch, P.A., Discriminant Analysis, Hafner Press, 1975

LaRue, M.A., J.J. Rotella, R.A. Garrott, D.B. Siniff, D.G. Ainley, G.E. Stauffer, C.C. Porter, P.J. Morin, "Satellite imagery can be used to detect variation in abundance of Weddell seals (Leptonychotes weddellii) in Erebus Bay, Antartica," Polar Biol., 34, 1727-1737, 2011.

McMahon, C.R., H. Howe, J. van den Hoff, R. Alderman, H. Brolsma, M.A. Hindell, "Satellite, the all-seeing eyes in the sky: counting elephant seals from space," PLoS ONE, 9(3), doi:10.1371/journal.pone.0092613, 2014.

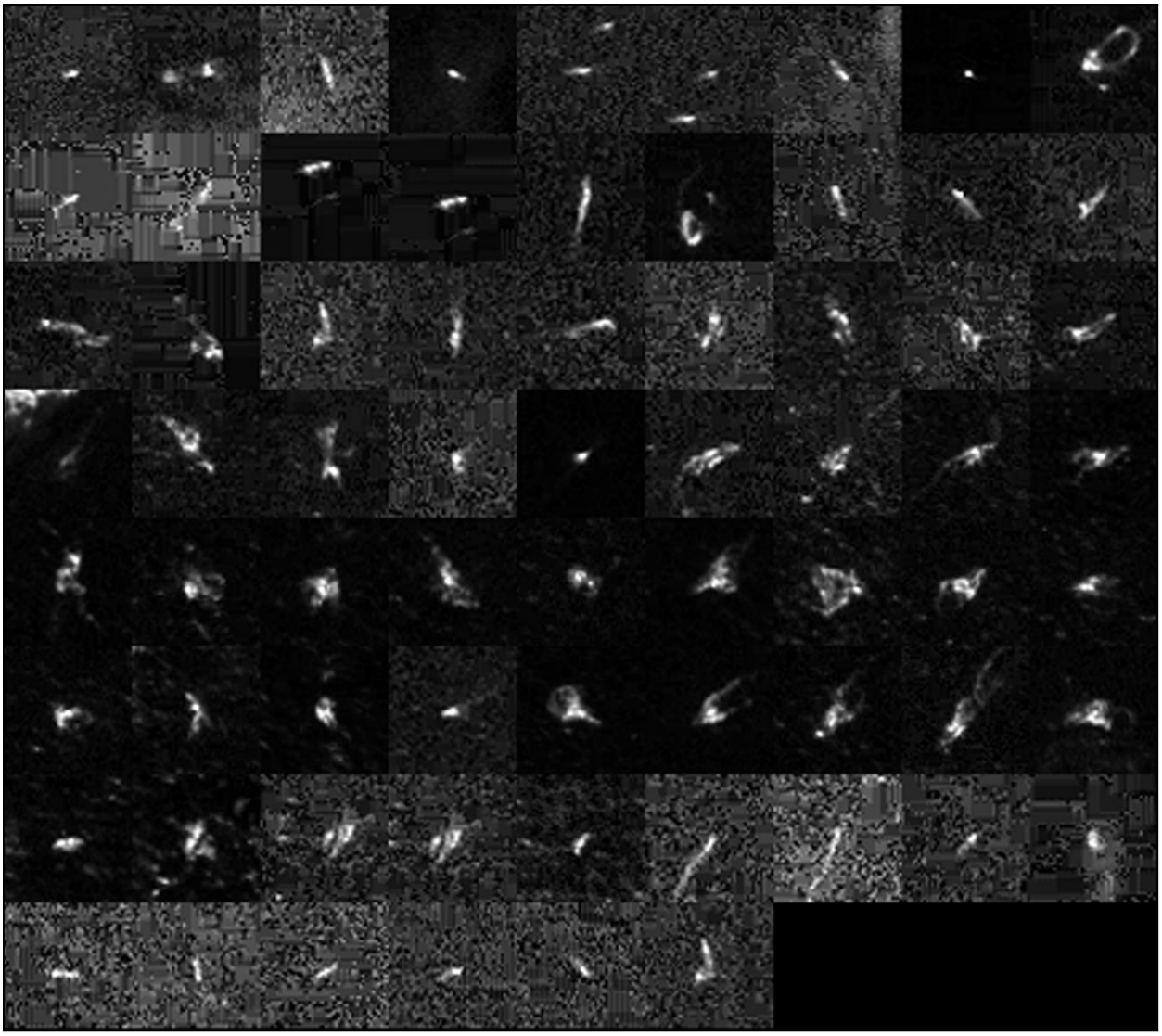
Stapleton, S., M. LaRue, N. Lecomte, S. Atkinson, D. Garshelis, C. Porter, T. Atwood, "Polar bears from space: assessing satellite imagery as a tool to track arctic wildlife," PLoS ONE, 9(7), doe:10.1371/journal.pone.0101513, 2014.

Wackerman, C., W.G. Pichel, P. Clemente-Colon, "Automated location of ice regions in RADARSAT SAR imagery," Coastal and Marine Applications of SAR, ESA SP-565, p.169-174, June 2004

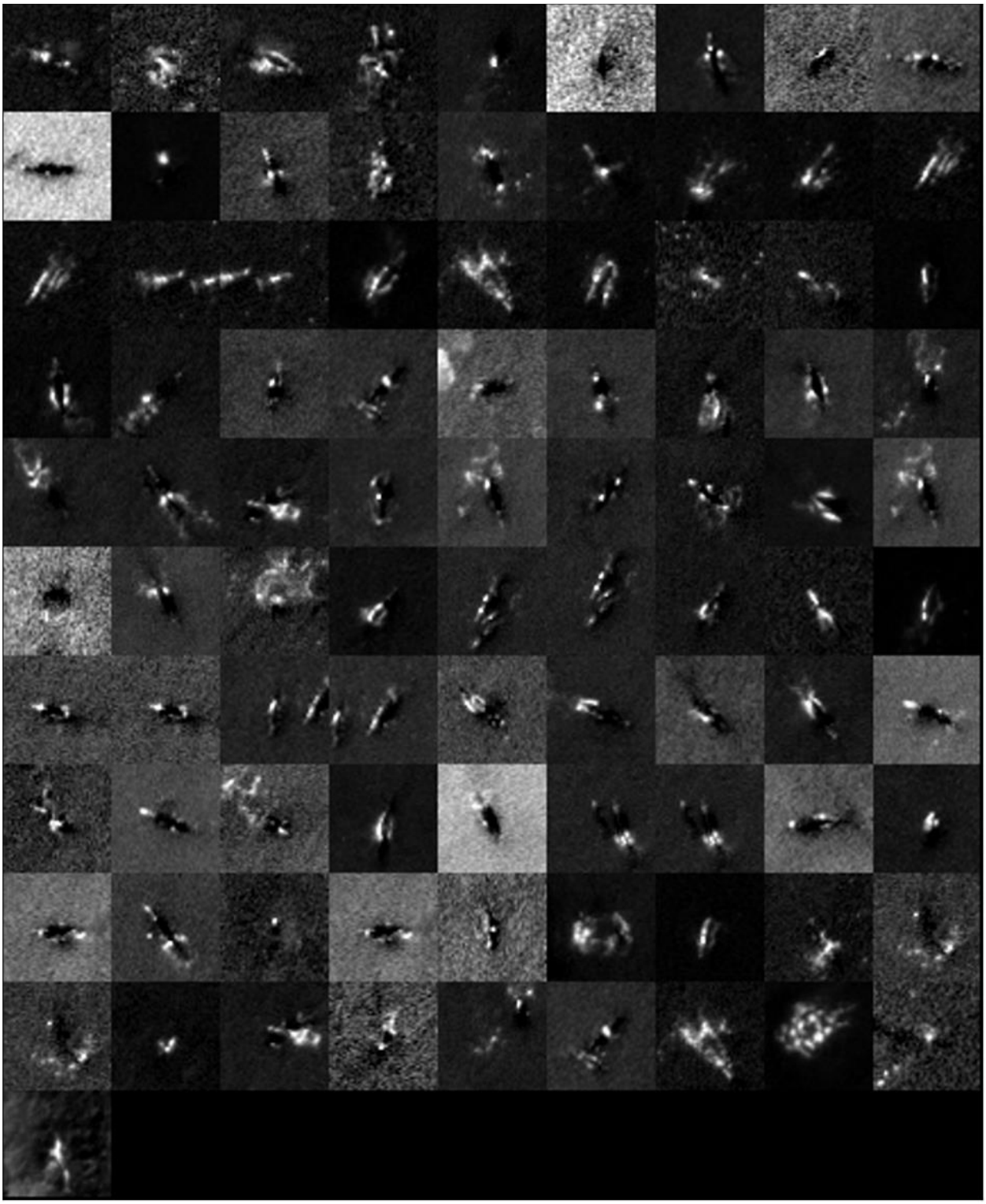
Wackerman, C., Miller, D. An Automated Algorithm For Sea Ice Classification In The Marginal Ice Zone Using ERS-1 Synthetic Aperture Radar, *Erim Technical Report 252000-25-T*, May 1996.

Wackerman, C. Use of An Interferometric SAR For Terrain Classification, *Proc. 26th AIPR Workshop*, SPIE vol. 3240, 75-86, 1997.

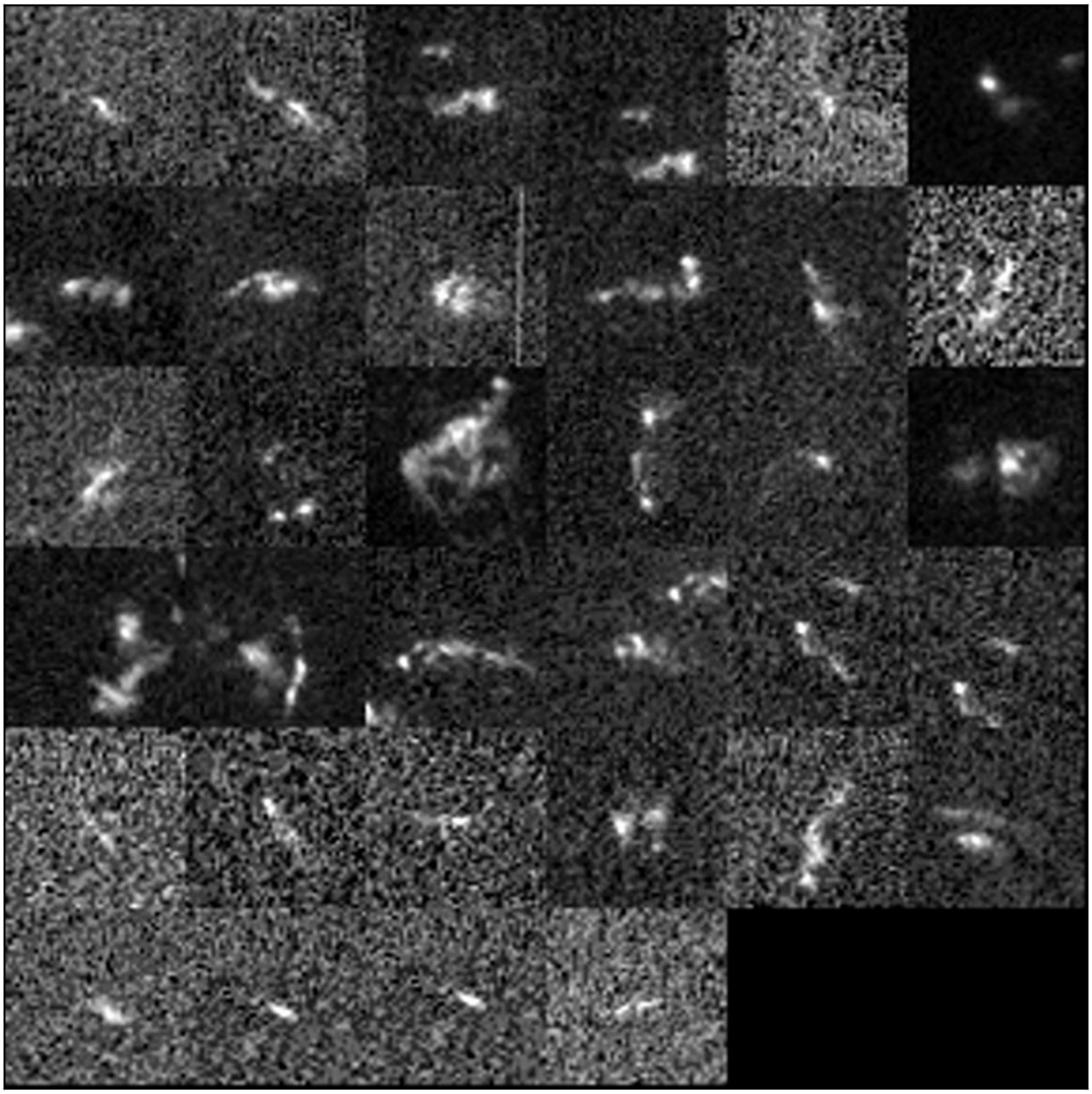
Appendix A: Images for all the whale signatures extracted for this study.



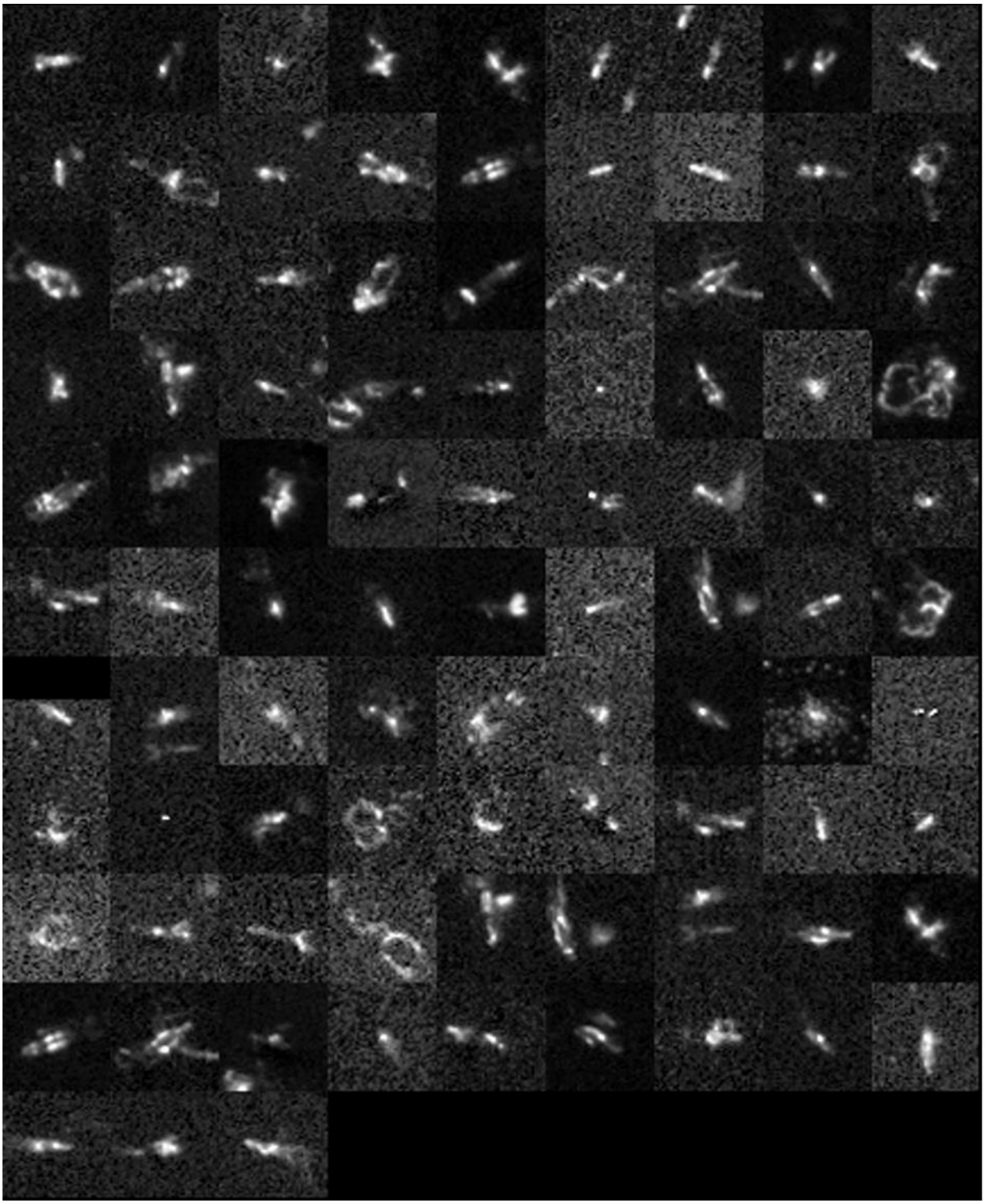
Auckland 2006 Pan whale signatures; QuickBird-2 resolution=0.66 m



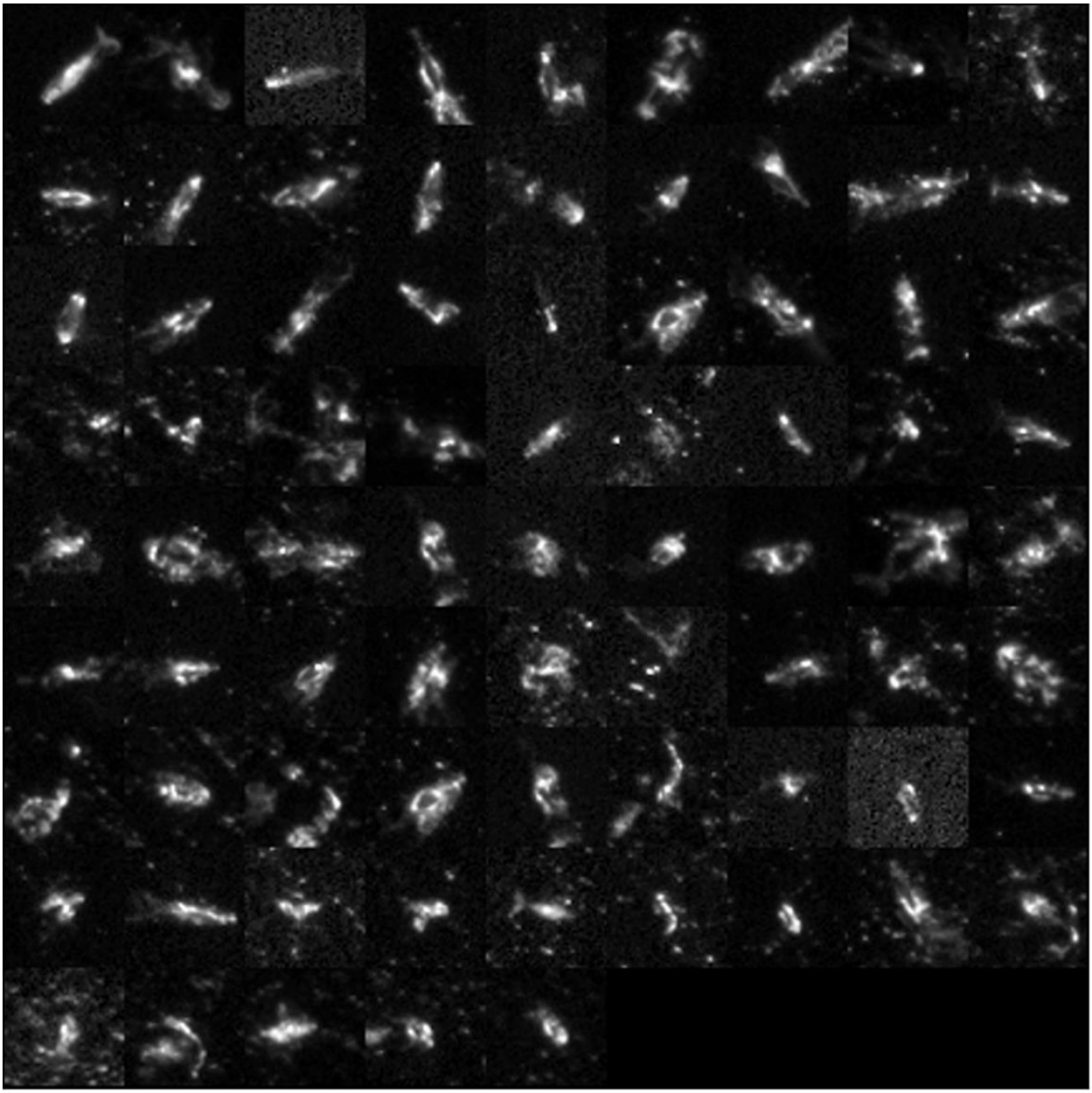
Witsand 2009 Pan whale signatures; GeoEye-1 resolution = 0.43m



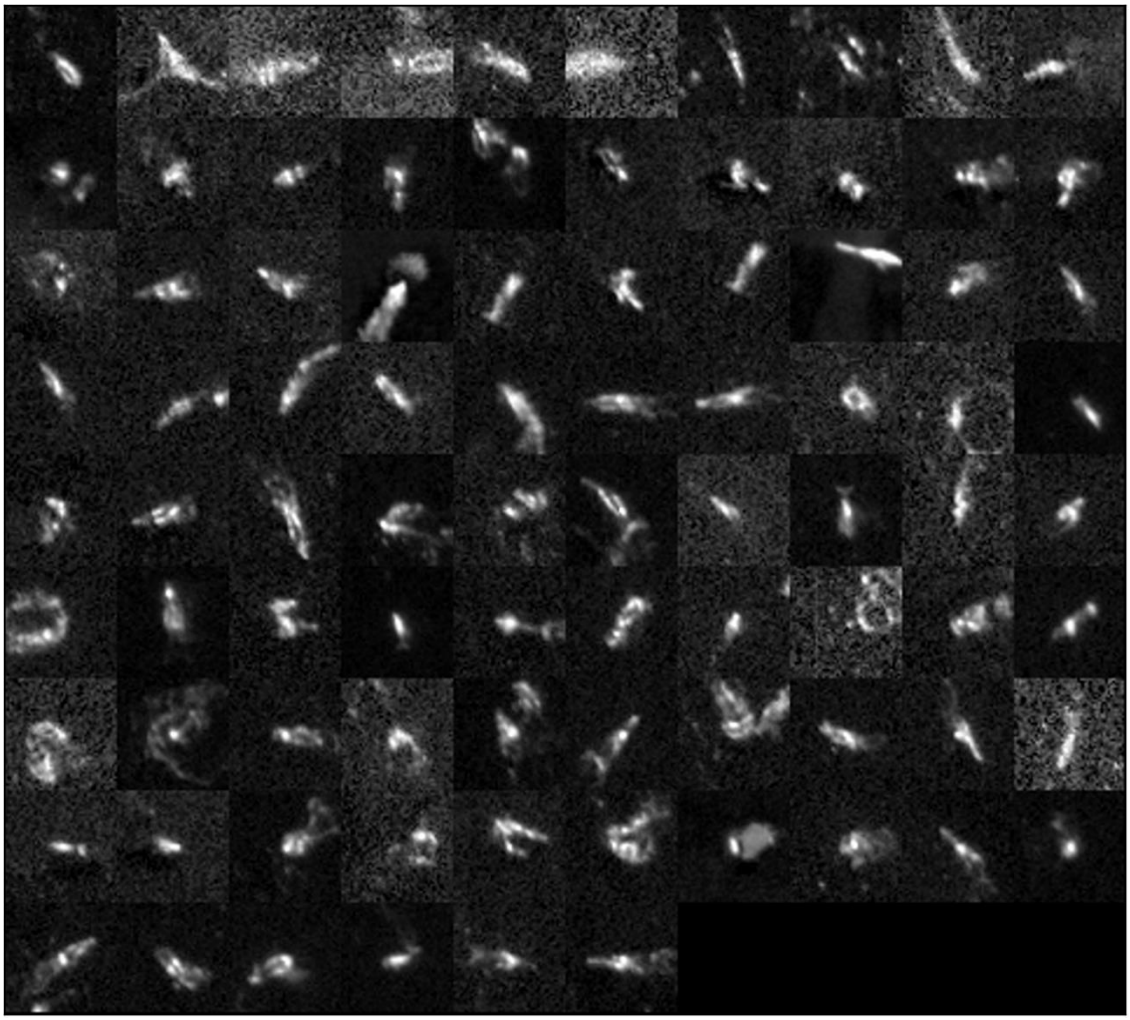
Auckland 2011 Pan whale signatures; WorldView-2 resolution=0.47m



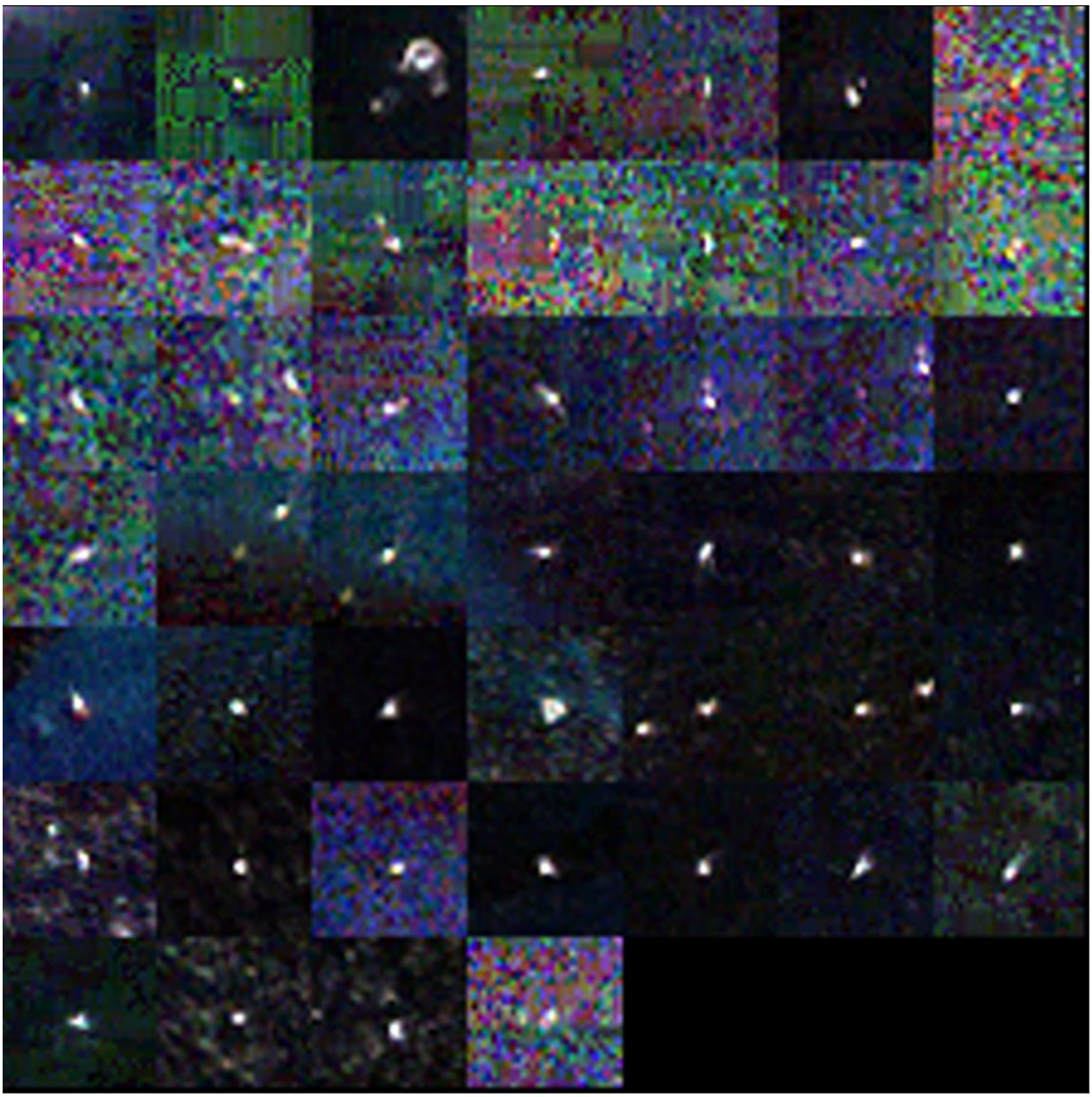
Valdes 2012 Pan whale signatures; WorldView-2 resolution=0.52m



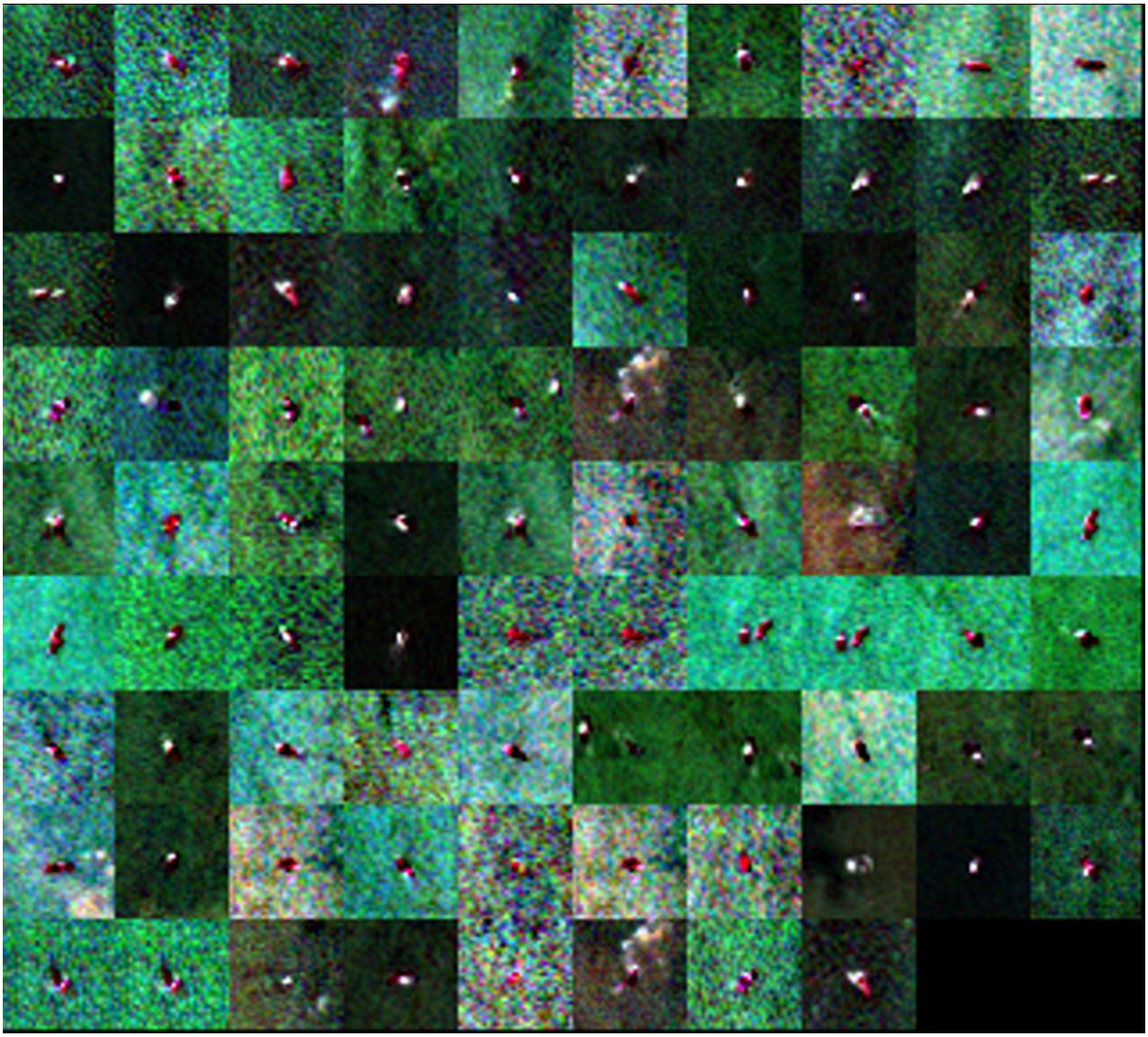
Valdes 2014 Pan whale signatures; WorldView-3 resolution = 0.36m



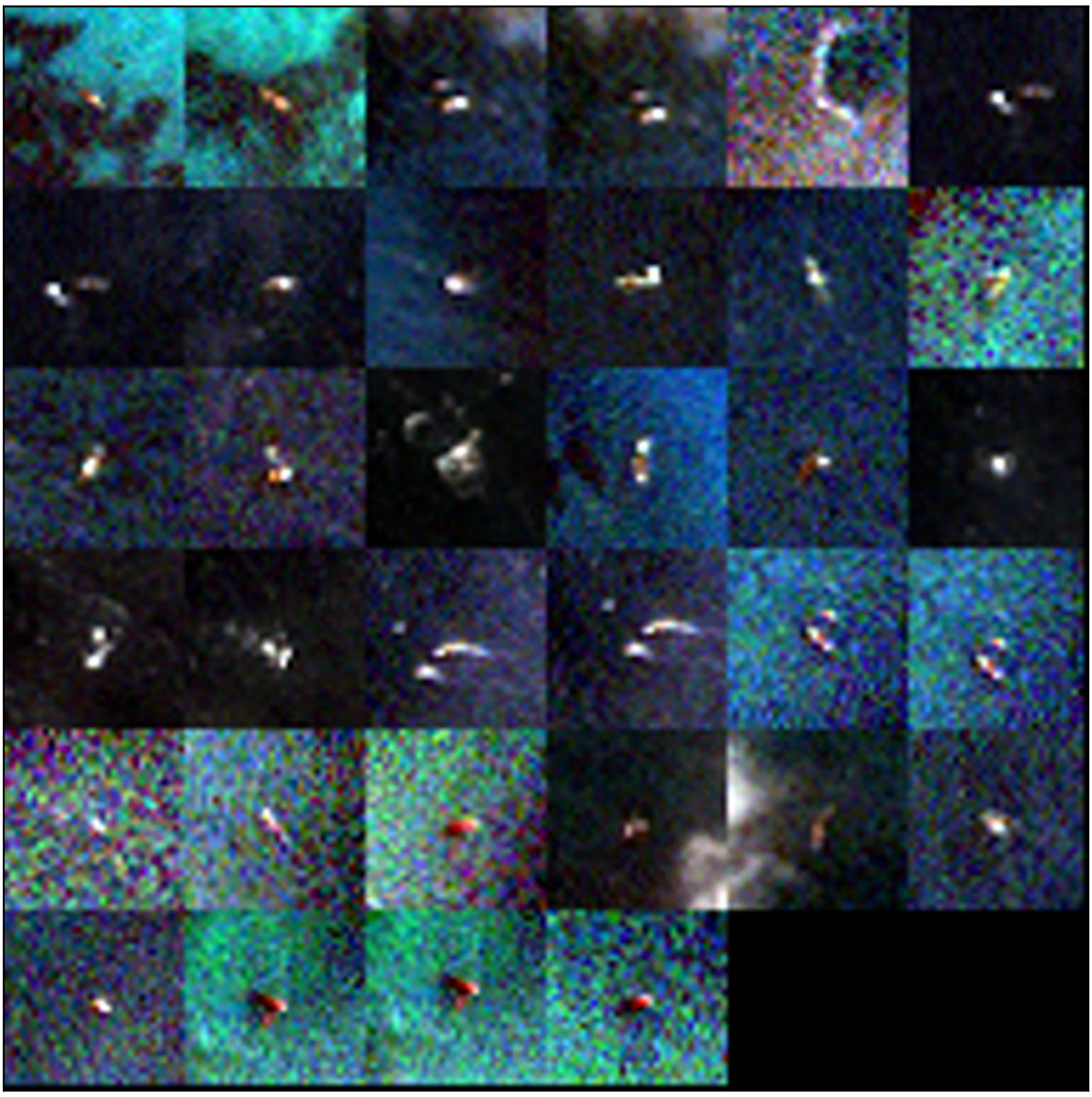
Valdes 2016 Pan whale signatures: WorldView-2 resolution=0.54m



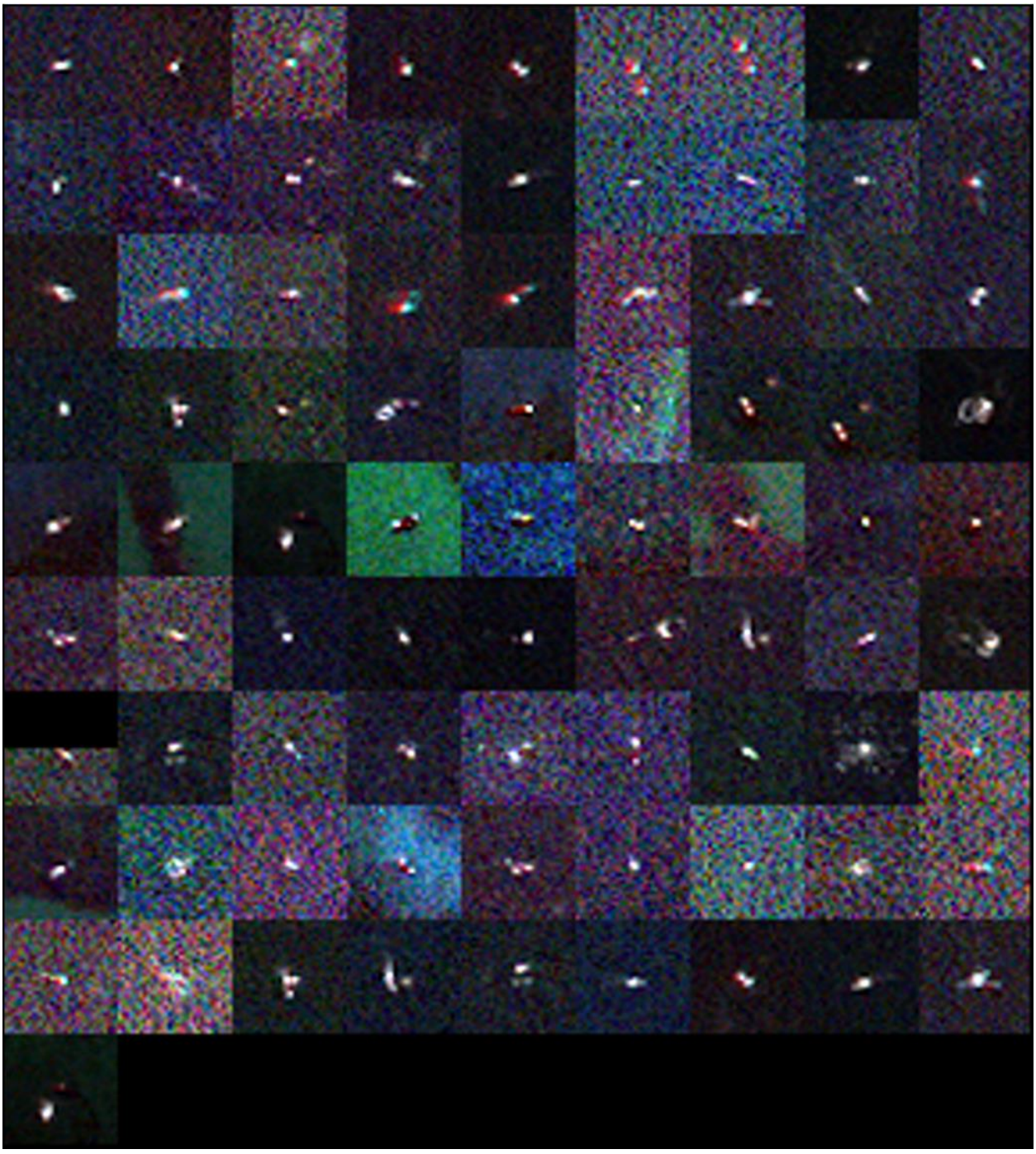
Auckland 2006 MSI whale signatures; QuickBird-2 resolution=2.6m



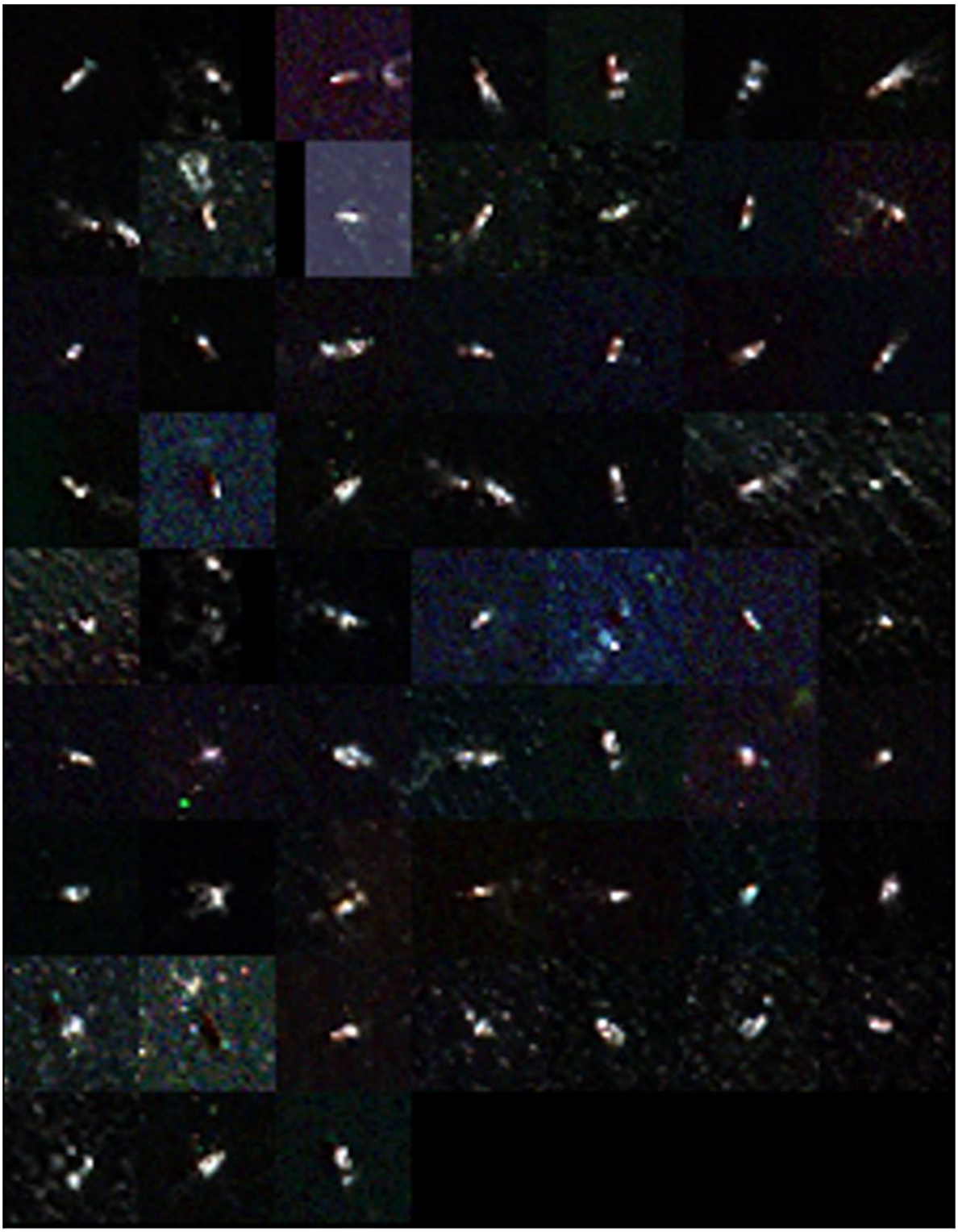
Witsand 2009 MSI whale signatures; GeoEye-1 resolution=1.7m



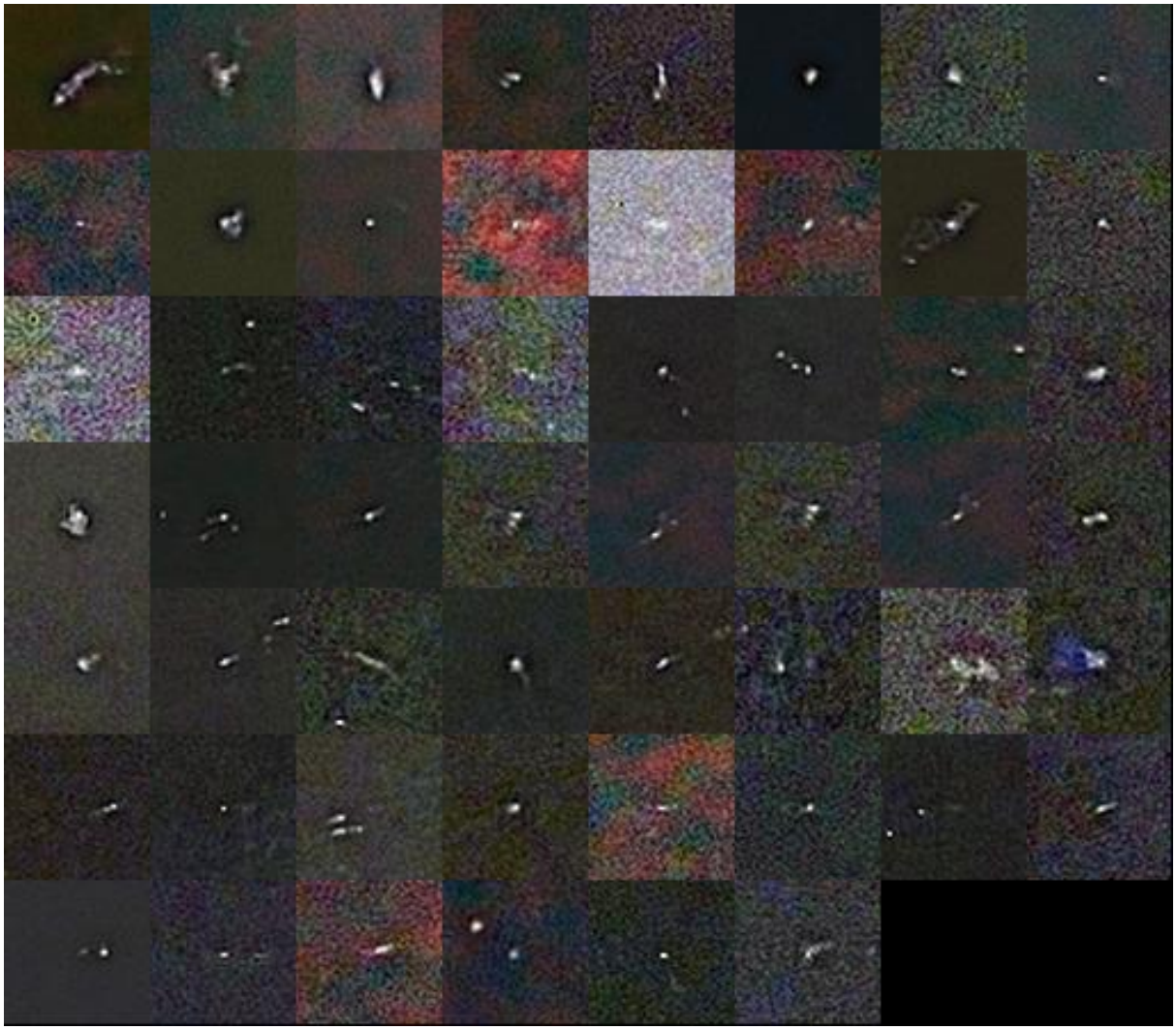
Auckland 2011 MSI whale signatures: WorldView-2 resolution=1.9m



Valdes 2012 MSI whale signatures; WorldView-2 resolution=2.1m



Valdes 2014 MSI whale signatures; WordlView-3 resolution=1.4m



Cape Cod 2021; Pleiades resolution=0.73m