Naval Research Laboratory
Washington, DC 20375-5320

NRL/7350/MR—2023/2

Improved Automated Whale Signature Detection with Combined PAN and MSI Feature Vectors

CHRISTOPHER WACKERMAN

Seafloor Sciences Branch
Ocean Sciences Division

August 6, 2023

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*
06-08-2023	NRL Memorandum Report	01/04/2021 – 31/12/2022

4. TITLE AND SUBTITLE

Improved Automated Whale Signature Detection with Combined PAN and MSI Feature Vectors

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--2023/2

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

Estimation of improvement to automated whale signature detection using panchromatic and multi-spectral images when the two modalities area combined into a single feature vector.

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**	U	31	**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U			(228) 688-5354

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

This page intentionally left blank.

Improved Automated Whale Signature Detection with Combined PAN and MSI Feature Vectors

1.0 Summary

The overall goal of the current tasks on this project were to start to move to development of a useful detection capability that could be used operationally on WorldView imagery. What this meant was essentially to start working on mitigating false alarms in imagery that was not included in the training set so that we had a useful tool for extracting whale signatures from new imagery. We were able to develop new detection capabilities that now can run on an entire WorldView Ortho or Basic image and generate a much smaller number of detections that can easily be manually reviewed yet still maintained a high probably of detecting actual whale signatures. We now have a good start to an actual capability for a user to realistically review imagery to find new whale signatures.

The following improvements were made to the detection algorithms.

- Many of the false alarms were from breaking wave signatures that generated a larger number of bright "blobs" throughout the image chip. Since whale signatures have more concentrated bright regions, we added a new feature to the feature vector that measured the distances between all pairs of local maximum in the panchromatic (PAN) image and recorded the maximum distance and the standard deviation of the distances.

- We added a capability to generate a single feature vector with metrics from both PAN and multispectral (MSI) imagery. The user supplies the image name for the PAN image, and the code now automatically finds the corresponding MSI image and finds the image pixels that correspond to the PAN image chip. We then create a feature vector that has the set of PAN features (with an option of normalizing the image statistics by dividing by the mean) plus the mean values of all the MSI bands over the target box. We also put in the option of normalizing the MSI band means by dividing each band mean by the Coastal band mean.

- We added 400 background examples into the training set from the Maine imagery to try and add more/different breaking wave signatures into the training set.

The performance of the new detection algorithms were generated by testing them on the training set, and on a test set of full imagery that consisted of: (1) 4 Worldview Ortho images from the Valdes 2012 collection and 4 Worldview Ortho images from the Valdes 2014 collection with manually-derived whale signature; and (2) 6 Worldview Basic images collected off the coast of Maine. The neural net algorithm outperformed the optimal linear combination algorithm, and the best performance was generated using non-normalized MSI imagery with either normalized or non-normalized PAN imagery (it did not matter which we used). A summary of the performance of the neural net algorithm using normalized PAN and non-normalized MSI with a detection metric threshold of 0.5 follows.

Manuscript approved July 11, 2023.

- On the training set of Worldview images consisting of 9340 background chips and 570 whale signature chips, the average over different trials of the number of false alarms (NFA) was 10, which generates a probability of false alarm (PFA) of 0.0010 and a number of false alarms per Worldview Ortho Tile (NFA/Tile) of 67. Over different trials these numbers varied as: NFA 7-16; PFA 0.0007-0.002; NFA/Tile 49 - 112
- On the training set, the number of missed detections (NMD) on the average was 24, generating a PD of 0.96. Over different trials these numbers varied as: NMD 11-29; PD 0.95-0.98
- On the Valdes imagery, the NFA/Tile varied from 4-499 with an average of 124. On the Maine data the NFA/Tile varied from 0-844 with an average of 276. The largest values of NFA/Tile came from imagery with very strong waves and breaking waves.
- On the Valdes imagery the PD varied from .80 – 1.0 with an average of 0.94
- Averaging training set and full imagery results, the new detection algorithm achieves a **PD of ~0.95 with NFA/Tile of ~157. We are now well within the range of a realistic operational tool for generating possible whale signatures from full image for review.**

The following new applications were developed.

- Stand-alone routine to read MAXAR PAN and MSI imagery without using any Python packages. Previously we used the Python gdal package to read the imagery, but when our laptop was upgraded to the next Python version, gdal no longer worked. This appears to be a well known problem in the community that gdal is notoriously difficult to import. None of the other Python TIF readers worked on the MAXAR MSI imagery. So we have developed our own reader that does not require any TIF Python packages. It has been tested against gdal (using our old laptop) for all the different image types and reproduces the same images exactly.

- Automated capability to find the MSI pixels that correspond to associated PAN pixels. This was required in order to be able to have both PAN and MSI feature metrics in the same feature vector. This also now allows us to generate pan-sharped color images of the whale/background signatures. Three pan-sharpened algorithms have been incorporated into the display utility: Brovey, ESRI, Simple Mean.

- Better tool for providing detections for review that displays pan-sharped image chips (making it easier to distinguish surface boats and land features) and sorts the detections based on the detector metric so that the detections with the highest probability of being a whale signature are shown to the user first.

- A tool to automatically generate a land/water mask for each image that uses the TOPEX worldwide bathymetry database combined with an image threshold operation. This is an automated process, so we now generate a water mask for every image before applying the detection algorithms which has significantly decreased the number of land detections.

- Tool to perform Principle Components Analysis (PCA) on the feature vectors in the training set. This allows us to get a better understanding of the "shape" of the feature

space, what the true dimensionality is, and which features dominant the target and background space.

- Ability to perform the anomaly filter after running the detector. The detector now records the anomaly statistics for every application of the detector (not just the detections) so that after running the detectors the full anomaly filter code can then be run.

2.0 Improvements to the Detection Algorithms

In this section we will provide more details about the various improvements that were made to the detection algorithms

Summary of Detection Algorithms

As noted in a number of previous reports, we have two detection algorithms (see [1] through [5] for details). For both of these, we move a local window through the image. For each placement of the window we calculate the feature vector. We then want to determine if this feature vector comes from a whale signature or a background signature. This is where the two algorithms differ. One algorithm, referred to as OptLin, derives the optimal linear combination of features vector values to separate target vectors from background vectors using the training set. The linear combination can be defined by a classification vector that the image feature vectors are projected onto, then the value of this projection (i.e. where along the classification vector the feature vector is projected to) is thresholded to determine if it is a target (greater than the threshold) or a background (less than the threshold) image chip. The classification vector is optimal in the sense that it maximizes a metric known as the Fischer Discriminant which is defined as:

$$(\text{targ-mean} - \text{back-mean})^2/(\text{targ-var} + \text{back-var})$$

where targ-mean, targ-var are the mean and variance, respectively, of the target projection values and back-mean, back-var are the mean and variance of the background projection values. We normalize these projection values such that for the background vectors their mean is 0 and for the signal vectors their mean is 1.

The second algorithm finds a neural net that separates the target vectors from the background vectors. We use a net that generates a number that is then thresholded to determine if it is a target vector (greater than the threshold) or a background vector (less than the threshold). The neural net uses the training set to derive the parameters such that the mean-squared-error between the neural net output value and a value of 0 for all the background vectors in the training set and 1 for all the target vectors is minimized. In other words, the net is trained to try and generate a 0 if it is a background vector and a 1 if it is a target vector.

For each algorithm, we can get a more robust estimate of its performance by performing a number of trials, where for each trial we randomly divide the training set into training vectors and testing vectors and generate detection perform for each trial. For the OptLin algorithm we generate a new classification vector in each trial so we can then generate a classification vector

that is the average of all of the classification vectors generated at each trial. The idea here is that the averaged vector may be less sensitive to the statistical details of the individual training/testing set distributions. For the neural net algorithm, each trial (with a different set of training/testing feature vectors) is actually used to improve the existing neural net. Thus the final neural net is in a sense the equivalent of the averaged classification vector in OptLin. Thus we actually have three detection algorithms: OptLin (where the classification vector is generated from the entire training set); OptLin Avg (where the classification vector is generated as the average of the vectors from each trial); and NNMSE which generates the neural net using all the trials.

New PAN Feature

One of the major false alarms that we see in the results comes from breaking wave signatures in the PAN image. One feature of many of these image chips is the existence of multiple bright "dots" throughout the chip. Whale signatures, on the other hand, tend to have their bright response clustered in a much smaller area. To try and capture this, we added two new features to the PAN feature vector. For a given image chips, we locate all of the local maxima defined as an image pixel that is within 0.5 of the global maximum and is the largest values within a 11 x 11 region centered on the pixel. We then calculate the distance (in pixels) between all pairs of these local maxima. The two metrics are then the maximum value of these distances and the standard deviation. We show below how adding these features improved detectability slightly.

Details on the full set of PAN features can be found in previous reports (see [3] and [4]). The full set now consists of the following 16 (or 15 if we are normalizing the statistics) entries:

4 anomaly metrics defined as (see [3] for details):

- mnmet = $(m_T-m_B)/\sigma_B$ where m_T is the mean value within the target box, m_B is the mean value within the background ring, and σ_B is the standard deviation within the background ring (see [1] and [2] for definitions of these various boxes);
- varmet = σ_T/σ_B where σ_T is the standard deviation of the target box (note that this is slightly different than how we defined it in [3]);
- minmet = the number of pixels in the target box that are smaller than they should be based on a Gaussian distribution with statistics m_B, σ_B ([3] has details on how to implement this);
- maxmet = the number of pixels in the target box that are larger than they should be based on a Gaussian distribution with statistics m_B, σ_B ([3] has details on how to implement this).

6 statistics defined as:

- m_T
- sd = σ_T
- m3 = E[x**3]**0.3333
- m4 = E[x**4]**0.25
- s3 = E[(x-m_T)**3]**0.3333
- s4 = E[(m-m_T)**4]**0.25

where E[] is the expected value of what is inside the brackets, and x represent pixel values within the inner target box. If we normalize the statistics (to make them scale factor invariant) we divide them all by the mean in the target box to generate 5 statistics (we lose the first one since dividing by it just makes it a constant of 1):

- sdnorm = σ_T/m_T
- m3norm = E[x**3]**0.3333/m_T
- m4norm = E[x**4]**0.25/m_T
- s3norm = E[(x-m_T)**3]**0.3333/m_T
- s4norm = E[(m-m_T)**4]**0.25/m_T

6 image metrics (these are the metrics we created specifically for whale signatures) defined as:

- 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 box
- delmax = 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.
- locmaxdist = the maximum distance between all pairs of local maxima.
- locsddist = the standard deviation of the distances between all pairs of local maxima

Note that if we normalize the statistics, all 15 metrics are scale-factor invariant.

Combining PAN and MSI features

This is the most significant improvement. We worked out the algorithm for finding the MSI pixels that correspond to a given PAN pixel. The MSI pixels are four times larger (in terms of spacing in meters) than the PAN pixels, so the MSI imagery has approximately ¼ x ¼ of the pixels that the PAN images have. For the WorldView Ortho imagery this mapping was trivial; the first MSI pixel maps to the first 4x4 set of PAN pixels; the second MSI pixels maps to the next 4x4 PAN pixels; etc. For the WorldView Basic imagery, the mapping is a little trickier since the MSI Basic imagery have additional pixels along each column than what can be accounted for from the PAN pixels. We have to find the number if these "padded" pixels in each line, and then add an offset in columns that is half of these pixels. In essence these "padded" pixels appear to be evenly distributed at the beginning and end of each line. This was all tested by looking at the PAN and MSI image chips near coastal features and for known whale signatures to make sure that the brighter MSI band values mapped to the bright PAN pixels.

With this capability we can now generate feature vectors that contain metrics from both the PAN and MSI imagery. The PAN metrics were listed above. For the MSI metrics, we note that the MSI image resolution is ¼ of what the PAN resolution is, so most of the MSI whale signatures turn out to be just single bright "dots" (see [1] and [2] for a full set of MSI whale signature image chips). Thus the image metrics used for the PAN imagery is not useful for MSI, and the anomaly statistics would just be a repeat of the PAN anomaly statistics. What the MSI does provide that the PAN does not have are the band spectral values. So for the MSI metrics, we used the mean

values of each band within the target box. Since WorldView imagery have 8 bands, this generated 8 additional metrics. As with the PAN data, we also want to have the option of making these metrics scale factor invariant, so there is the option of dividing each band mean by the mean of the first band (the Coastal band) which generates 7 additional metrics.

We looked at taking the mean of the MSI bands in a small region (3x3 or 5x5) around the peak of the sum of the Blue, Green, and Red bands in order to just get the spectral distribution of the location with the most energy, but this performed worse than just using the mean over the target box.

We note that since we are using the mean over a range of pixels, the mapping between PAN and MSI pixels could be off by a few pixels without having a significant effect.

Combining Detectors with Anomaly Filter

Because we are getting better results with the new detectors (see performance results in the next two sections), we now would use the detectors as the first attempt to find whale signatures in an image. However, it still is possible that we get too many detections to review, so we have built into the detectors the ability to perform the anomaly filter after the fact to weed down the number of detections to review. Details on the anomaly filter can be found in [3]. Since we have include the anomaly filter metric into the feature vector, we actually are calculating these values for each placement of the detection window. So what we do now is save these values in an image for each detection window placement. Then after running the detectors, we have the ability to use these output image data values and perform the anomaly filter as described in [3]. We also save the detection metric in an image file, so we have the added advantage of having that value also. To review the detections, we can also now sort them based on one of the metrics (either one of the anomaly statistics or the detection metric) so that we present to the user image chips to review first that are higher up on the probability of being a target. As with the original anomaly filter, the user then selects which detections to keep.

3.0 Performance on the Training Set

As with previous reports, we now present the performance of the detectors when applied to the training/testing set. This is a set of feature vectors that have been calculated from manually identified whale signatures and background signatures; see [1] and [2] for a full description of how these chips were identified and example images of the signature. For the results in this report we limited the training/testing set to Worldview Basic and Ortho images (so did not include the GeoEye or QuickBird examples that are in the set) so that we always had 8 MSI bands available. This training set has 9340 background feature vectors and 570 whale feature vectors.

Effect on Detectability of the New Metrics

To see if the new metrics improved detection of the whale signatures, we looked at a range of detectability metrics using the different set of features

Both detection algorithms generate a number for each feature vector that is then thresholded to determine what class the vector is in. For the OptLin algorithm the number is the normalized projection along the classification vector. For the NNMSE algorithm the number is the output of the neural net. Thus we can assign a number to each of the whale vectors in the training set and each of the background vectors. We can then use these numbers to generate the Fischer Discriminant (defined above) for the given detector algorithm as well as the mean-squared-error (MSE) between the value and 0 for all background vectors and 1 for all signal vectors. This gives us two metrics for determining the detectability performance of the algorithms. Note that OptLin is designed to maximize the Fischer Discriminant over all linear combinations while NNMSE is designed to minimize the MSE.

There are other metrics that we can use. Because we are thresholding the output value to determine if the feature vector is a target or background vector, if we vary the threshold that we use then we vary the performance of the algorithm. We define the probability of detection (PD) as the probability of assigning a target vector to the target class. We define the probability of false alarm (PFA) as the probability of assigning a background vector to the target class. As we vary the threshold, we vary the resulting values for PFA and PD. Thus we can make a plot of PD versus PFA; this is referred to as a Receiver Operation Characteristic Curve (ROCC). As we change the value of the threshold the values of PFA go from 0 to 1 and the values of PD go from 0 to 1. All ROCCs go from (0,0) to (1,1) since for a threshold larger than all the values we have PD=PFA=0 and for a threshold smaller than all the values we have PD=PFA=1. A ROCC has to be above the 45 degree line, since if it is below it than that means that the background values are larger than the target values, and we could just switch the direction of the threshold and flip the curve about the 45 degree line. A measure of how good a ROCC is, is how much it is squished into the upper, left corner where PD=1 and PFA=0. A perfect detection algorithm would have PD=1 for all values of PFA. Thus two additional measure of detection performance are: (1) the area under the ROCC, since as the ROCC pushes toward the upper, left corner the area goes to 1; and (2) 1.0 – (the shortest distance of any point on the ROCC to the PD=1, PFA=0 location). For perfect detection, both of these are 1. The smallest that the area metric can be is 0.5 (for the 45 degree line) and the smallest that the distance metric can be is 1.0-sqrt(1/2).

Finally we have used in past reports a metric which is the number of false alarms (NFA) that would be expected within a WorldView Ortho Tile if we set the PD to 0.90. A WorldView Ortho PAN Tile is 16384 x 16384 pixels. The NFA/Tile will be the PFA times the number of times we are applying the detection window to the tile. Since we move the detection window by the size of the guard box (see [1] and [2] for details), and the guard box is 64x64 pixels for the PAN imagery, the number of times we apply the detection window is (16384*16384)/(64*64) = 65536. So NFA/Tile at PD=0.90 is PFA*65536 where the PFA is from the ROCC when PD=0.90. This will be called “NFA/Tile PD 90”.

We note that all of these metrics are independent of the threshold that we use. They are measures of the separation of the whale and background feature vectors when using the two detection algorithms. However, when we actually apply the detection algorithm to an image we need to use a threshold. For a given threshold we then can have two additional metrics: the number of false alarms (NFA) and the number of missed detection (NMD). Note that these will change (usually significantly) as we change the threshold value. Note that we can translate NFA

to the expected number of false alarms within an Ortho Worldview tile as NFA/Tile = NFA*65536/9340. This is ***not*** normalized to PD=0.90, but is for whatever the PD that was generated using the algorithm threshold so it will change with a changing threshold. PD is then (570-NMD)/570 since there are 570 target feature vectors.

There are two main ways for choosing a threshold. One way is to find the threshold that minimizes the probability of an error. This is defined as the PFA plus the probability of a missed detection which is 1-PD. So we can find the threshold that generates the smallest PFA+1.0-PD. The other way is to note that both detection algorithms are generated output values that are meant to be 0 for a background feature vector and 1 for a target feature vector. If the "noise" is the same for both, then we should chose a threshold half-way between them, or at 0.5. We will show below that for OptLin, the threshold that minimizes the probability of an error is very close to 0.5. However for NNMSE it is not, and we discuss below why this is and which is the best threshold to use.

Algorithm	Features	Metrics							Thres=0.5	
		Fischer	MSE	ROCC Area	ROCC Dist	NFA/Tile PD 90	NMD	NFA	NMD	NFA
OptLin	Old PAN-N	6.8	0.045	0.980	0.925	702	42	190		
	PAN-N	7.0	0.048	0.983	0.927	646	42	186		
	PAN-N & MSI-N	10.1	0.039	0.995	0.965	302	17	172		
	PAN-N & MSI	9.9	0.038	0.994	0.966	344	13	241		
	PAN & MSI	11.9	0.034	0.997	0.964	274	15	232		
OptLin Avg	PAN-N & MSI-N	10.1	0.039	0.995	0.966	309	15	202		
	PAN-N & MSI	9.8	0.038	0.994	0.965	344	14	236		
	PAN & MSI	11.9	0.033	0.997	0.964	267	14	244		
NNMSE	PAN-N & MSI-N	9.8	0.0080	0.994	0.968	331	18	200	62	36
	PAN-N & MSI	29.3	0.0031	1.000	0.991	11	3	71	24	10
	PAN & MSI	28.3	0.0030	1.000	0.990	10	3	76	20	12

Table 1: Detection metric results for the training set using different detection algorithms and different normalizations

Table 1 shows the various metrics for the various detection algorithms. The left-most column lists the detection algorithm: either OptLin, OptLinAvg, or NNMSE. The second column, labelled "Features" determines what the features in the feature vector are. There are the following:

Old PAN-N: This is the original PAN feature vector from previous reports, before we added the two additional metrics discussed in this report. The notation PAN-N means that we have normalized the PAN statistics. Similarly MSI-N will mean that we normalized the MSI band means. PAN, and MSI, without the –N means that no normalization has occurred.

PAN-N: This is the new set of PAN feature vectors discussed in this report, with the PAN statistics normalized.

PAN-N & MSI-N: This is the new feature vector combining PAN and MSI, with both normalized.

PAN-N & MSI: This is the new feature vector combining PAN and MSI with PAN normalized and MSI not normalized (so all 8 band means).

PAN & MSI: This is the new feature vector combining PAN and MSI with PAN normalized and MSI not normalized (so all 8 band means) and PAN not normalized (so all 16 features).

We then have range of detection metrics defined as:

Fischer: This is the Fischer Discriminant defined above.

MSE: this is the mean-squared-error between the detector output values and 0 for background vectors and 1 for target vectors.

ROCC Area: This is the area under the ROCC

ROCC Dist: this is the distance metrics for the ROCC defined above

NFA/Tile PD 90: This is the number of false alarms per Worldview Ortho PAN tile when PD is set to 0.90. This if for the threshold that minimizes the probability of error.

NMD: This is the number of missed detections from the entire training set. The first value for this uses the threshold that minimizes the probability of an error. The second value, shown only for NNMSE, uses a threshold of 0.5.

NFA: This is the number of false alarms from the entire training set. The first value for this uses the threshold that minimizes the probability of an error. The second value, shown only for NNMSE, uses a threshold of 0.5.

All of the results in Table 1 come from the training set. That is, these are the results of using the training set to generate the detection algorithm then testing it with data drawn from that set. In the next section we show results of applying the detection algorithms to full imagery. The boxes with a green background are the best metrics for that column.

It is important to note that the neural net algorithm generates a different neural net each time it is run, even if the set of feature vectors remains the same. This is not true of the OptLin algorithm. The optimal classification vector is determined by an analytical formula so it is the same for the same set of feature vectors. But the neural net will change, and that means the metrics in Table 1 will change with each run even when nothing else changes. So to deal with this the values in Table 1 for the NNMSE results are averages over five runs of the code. For anyone who is interested, Appendix A shows the individual values for each of the runs. Note that the OptLin Avg results will also change with each run because the target vectors used to create the classification vector will change randomly with each trial, thus generating a slightly different averaged classification vector each time. But we show below that the OptLin Avg performance is essentially the same as OptLin, so we just go with the OptLin classification vector.

Effect of New Feature Vector Metrics

The first three lines of Table 1 shows the detection metrics for the OptLin algorithm with the old PAN feature vector (top line), the PAN feature vector with the two new metrics discussed here (second line) and with the new combined PAN and MSI feature vector (third line). We can get a sense of whether this improvements actually did improve detectability by looking at the change in metrics across these algorithms. Comparing old-PAN-N versus PAN-N we get a slight improvement in all the metrics except MSE which went up slightly. So overall we get a slight improvement. Looking at PAN-N versus PAN-N & MSI-N we get improvement across all metrics, in many cases significant. So it seems that combining PAN and MSI into one feature vector is a significant improvement.

Effect of Normalizing PAN and MSI

For the three algorithms, we then have the metrics for the various combinations of PAN and MSI normalization. This is motivated in part by results presented in previous reports that indicated we got better detection results if we normalized the PAN data and if we did not normalize the MSI data. We did not do the PAN & MSI-N normalization since that appeared to be the worse of both worlds.

Looking at the OptLin results (lines 3,4,5 of Table 1), they are quite similar. PAN&MSI does give the best metrics for Fischer, MSE, ROCC Area, and NFA/Tile. PAN-N&MSI has the best ROCC Dist and lowest NMD while PAN-N&MSI-N generates the lowest NFA. OptLin Avg gives almost exactly the same trends. So for either OptLin algorithm, PAN&MSI is best followed closely by PAN-N&MSI. However NNMSE shows quite different trends (lines 9,10, 11 of Table 1). There is a huge difference between PAN-N&MSI-N and the other two, with the other two being significantly better. Between PAN-N&MSI, and PAN&MSI, it is a wash; all the metrics are essentially a tie.

So, putting all this together it seems that we are better not normalizing MSI (which is consistent with previous results) and it is kind of a wash whether we normalize PAN or not.

OptLin vs. NNMSE

Table 1 shows clear results that NNMSE performs significantly better than OptLin on the training set. This is also consistent with previous results (see [4]). Some insight into this can come from looking at the actual output values that are generated by the two algorithms. Figure 1 shows histograms of the target and background values for the OptLin (left column) normalized projection values and the NNMSE (right column) neural net outputs. The different lines correspond to the three normalization options, as indicated in the Figure. The purple vertical line

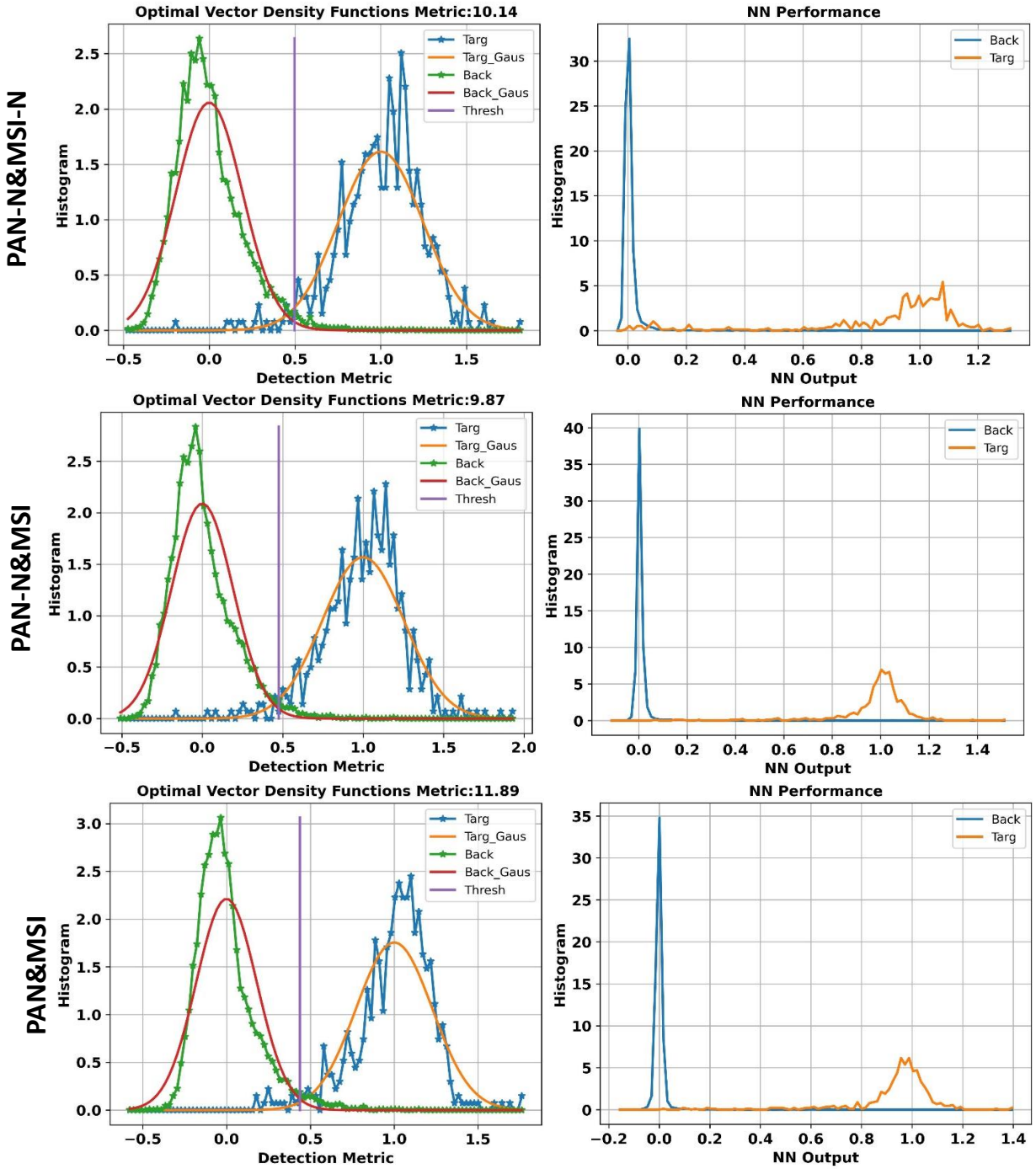

Figure 1: Histograms of the detection metrics from the OptLin algorithm (left column) and the NNMSE algorithm (right column) for the three different image feature normalizations (different lines). For OptLin, these are background and target histograms of the normalized projections onto the classification vector. For NNMSE these are histograms of the output of the neural net.

in the left plots indicates the threshold that minimizes the probability of error; by definition it will occur where the two histograms intersect. For the OptLin algorithm this threshold generally hovers around 0.5, indicating that the background and target histograms have similar widths (i.e. similar variances). The colored lines are these plots are a Gaussian density fit using the data statistics. The NNMSE results, on the other hand, show that the neural net bunches the background values into a very tight distribution around 0, but the target values are much more spread out, and in fact leak a bit into the background region. This difference in how the neural net treats background versus target vectors is most probably due to the different in the number of examples in the test set: 9340 for background vectors, 570 for targets. The threshold that minimize the error for these values hover around 0.1 (they range from 0.07 to 1.4 for the three sets of histograms). To see this effect more clearly, Figure 2 shows plots of the probability of error versus threshold value for the 6 cases. In Figure 2, the labels in these figures are slightly different than what was used in Table 1, so the translation is as follows:

WV_PanMsi_ntrials	=	OptLin PAN-N&MSI-N
WV_PanMsi_ntrials_nomsinorm	=	OptLin PAN-N&MSI
WV_PanMsi_ntrials_nonorm	=	OptLin PAN&MSI
WV_PanMsi_NNMSE	=	NNMSE PAN-N&MSI-N
WV_PanMsi_NNMSE_nomsinorm	=	NNMSE PAN-N&MSI
WV_PanMsi_NNMSE_nonorm	=	NNMSE PAN&MSI

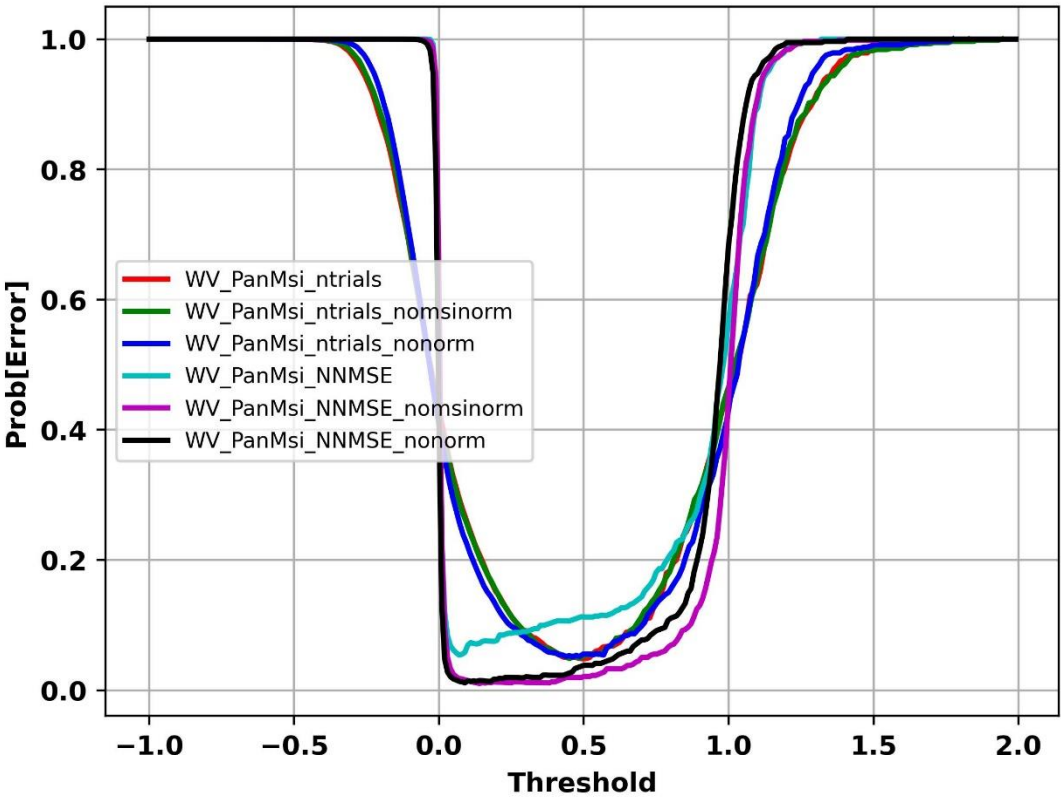

Figure 2: Probability of an error versus the threshold applied to the detector metric for OptLin and NNMSE detection algorithms for the three normalization options.

In Figure 2 all of the OptLin plots essentially lie on-top of each other and form very quadratic-looking curves with a minimum essentially at 0.5. Moving the threshold in either direction will quickly increase the probability of an error. For the NNMSE plot, the character is much different. There is essentially a region from about 0.1 to around 0.8 where we have a linear-like plot with a slight increase as we move to higher thresholds. In fact for NNMSE PAN-N&MSI (purple line) and NNMSE PAN&MSI (black line) the probability of an error is essentially flat until we get to around 0.5. This is why we want to look at two different thresholds for the NNMSE algorithm; one at the far left where the probability of an error is minimized, and one at 0.5 which is half-way between the background and target histogram peaks.

All of the results from Table 1 come from the ROCCs, so it is illuminating to look at them directly. The top plot in Figure 3 shows the final ROCCs for all of the algorithms. Note the limits on the axes; we are only showing the upper, left portion of each ROCC. The dotted lines are for the OptLin Avg results (the color is the same for the OptLin and OptLin Avg for the same normalization option). Note that all of the OptLin results (both OptLin and OptLin Avg) are essentially the same ROCC, indicating that it really does not matter which normalization we use, and whether we generate the classification vector from the full training set or use the average from the different trials. Note that this is consistent with the analysis of Table 1 that showed only slight differences in the detection metrics for the various normalizations.

As with the NNMSE Table 1 analysis, the ROCCs using non-normalized MSI are significantly better (pushed more toward the upper, left corner) than using the normalized MSI, and significantly better than the OptLin results. The NNMSE ROCC when normalizing both PAN and MSI (light blue line) is interesting because it is better than OptLin for very small values of PFA, but worse for larger values of PFA.

The bottom plot in Figure 3 shows all of the ROCCs from the trials. They all generally scatter around the corresponding ROCCs in the upper plot, but note that the OptLin results do show more scatter than the NNMSE results

Based on the results from the training set, the best detector algorithm is to use NNMSE with unnormalized MSI. Using a threshold of 0.5 this generates PD ~ 0.96, PFA ~0.001, and NFA/Tile ~75. Note that NFA/Tile at PD=0.90 is ~10. Looking at the variations in these numbers over the different runs we get: PD in [.95, .98], PFA in [.0007, .002], NFA/Tile in [49,112]. If we use the smaller threshold both PD and PFA go up (0.99 and 0.008 respectively), however we show below that using this lower threshold generates too many false alarms for full images.

Which Training Set Image Chips Cause Errors

It is useful to see which of the manually derived training image chips are causing the errors. Figure 4 shows the missed detection image chips using the OptLin algorithm for PAN&MSI (left image) and PAN-N&MSI (right image). The image chips in Figure 4 are sorted in decreasing order of the classification vector projection value, going from left to right along each line for each image. Thus the upper, left chip has the highest value (so closest to 1 and most like a

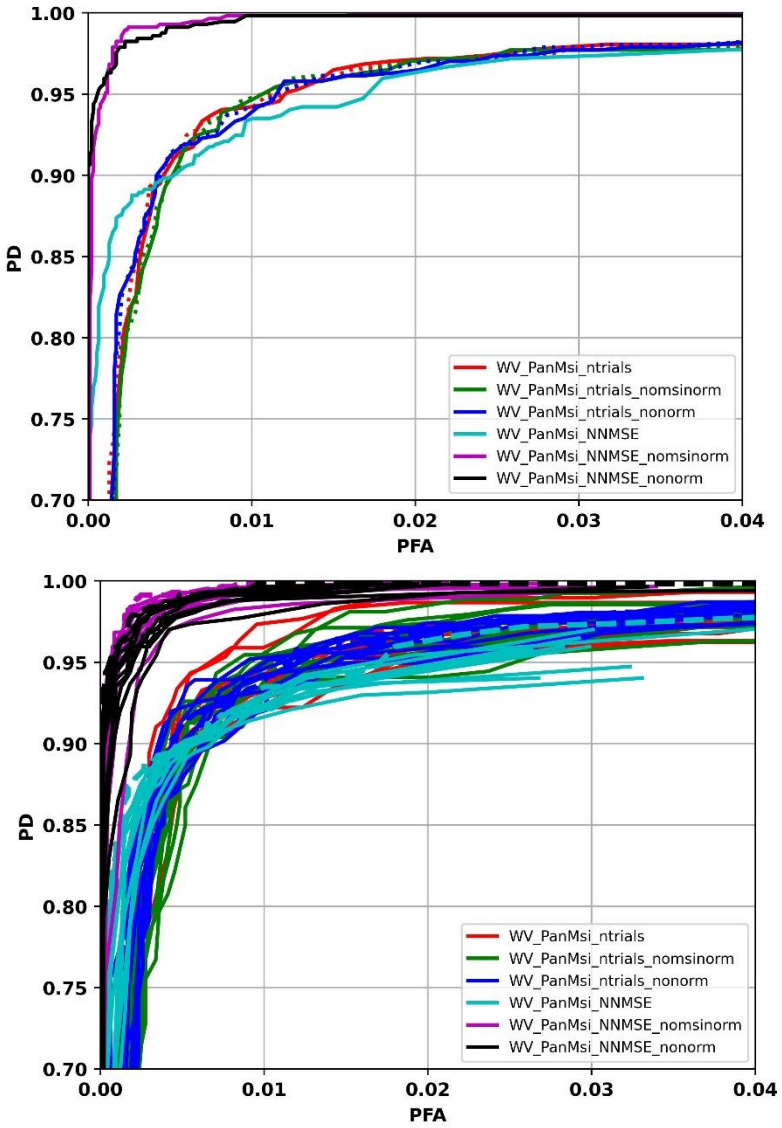

Figure 3: Top plot are ROCCs for the OptLin algorithm (labelled "WV_PanMsi_ntrials") and the NNMSE algorithm (labelled "WV_PanMsi_NNMSE") for the three different normalizations (no extension on label, extension "nonmsinorm", extension "nonorm"). Top plot shows average ROCCs over all trials; dotted lines are for OptLin Avg. Bottom plot shows the individual ROCCs for each trial, showing the spread in performance over trials.

target) whereas the lower, right chip has the lowest value (closest to 0 and most like a background). All but 2 missed detections in PAN&MSI are also in PAN-N&MSI, whereas the latter has an additional 5 chips. But they both contain very similar looking chips. Figure 5 shows the missed detections for the NNMSE algorithm for PAN-N&MSI; these are sorted in decreasing order of the neural net output. This is for one of the runs, there are a few chips that vary between the runs but overall they look very similar. There is significant overlap with the OptLin missed detection chips but also some additional chips. However overall the types of image chips that represent missed detections are very similar between the two algorithms. Note that in Table 1 OptLin has a smaller NMD than NNMSE (threshold 0.5), but larger NFAs.

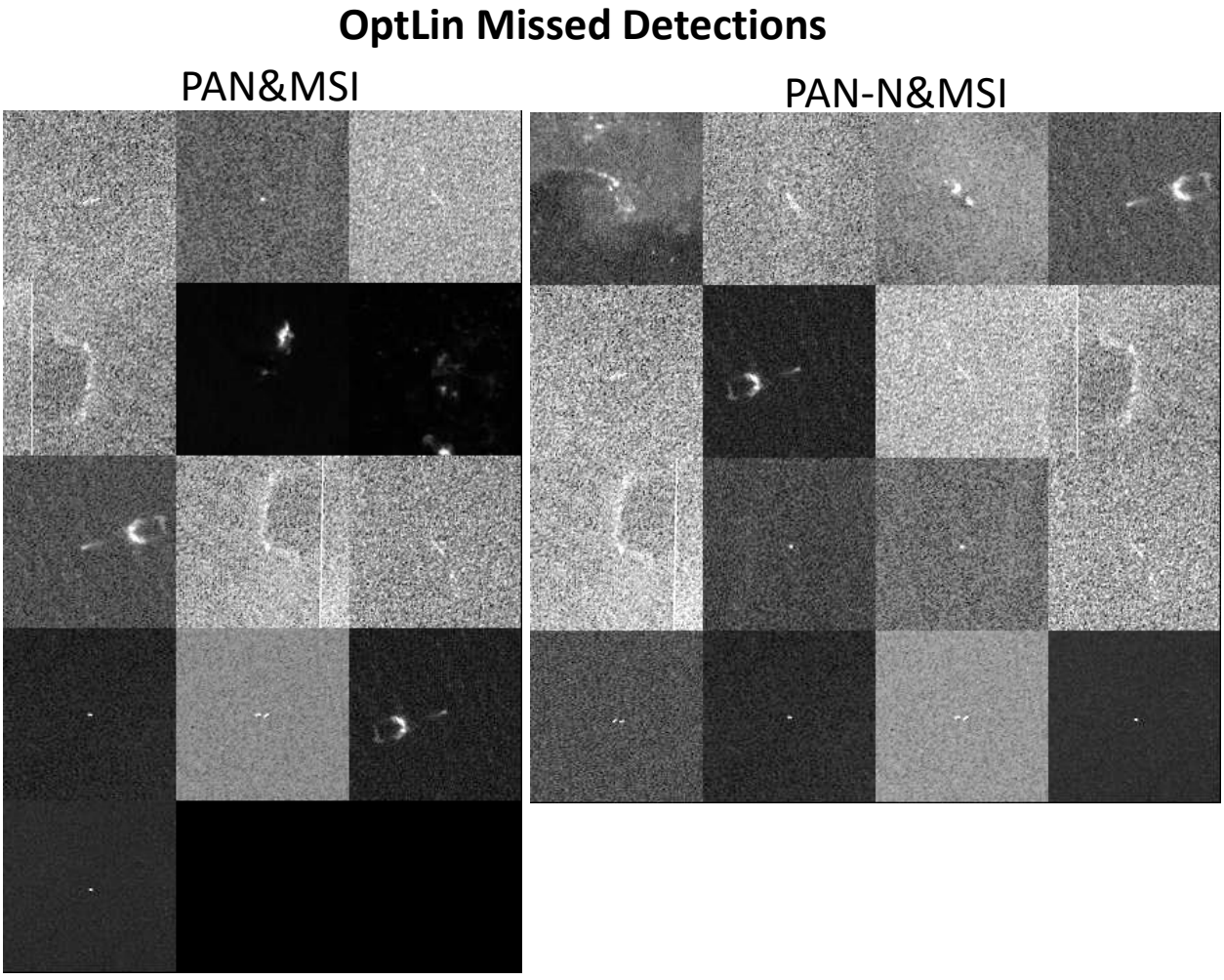

Figure 4: Missed detections from the training set using the OptLin detection algorithm for two different normalization options; PAN&MSI (left set of image chips), and PAN=N&MSI (right set of image chips).

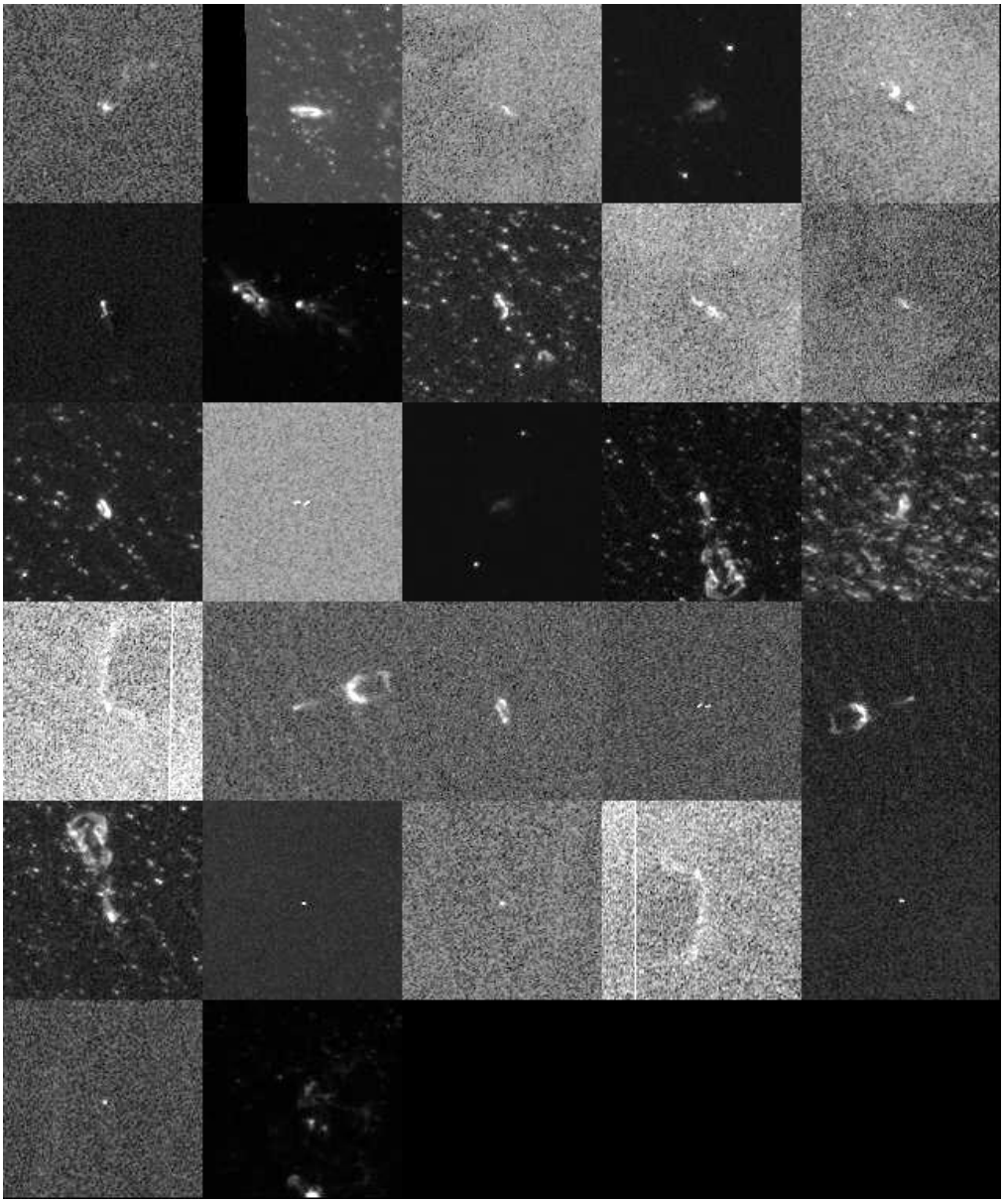

Figure 5: Missed detections from the training set using the NNMSE detection algorithm with a threshold of 0.5 for the PAN-N&MSI normalization option.

The missed detections between OptLin and NNMSE are very similar, however the false alarms are not. Figure 6 shows the first 100 false alarms from OptLin for PAN-N&MSI. These are sorted from left to right and top to bottom based on the classification vector projection value. Note that these correspond to bright center blobs as well as breaking waves. The next 106 are very similar. In contrast Figure 7 shows the NNMSE false alarms for PAN-N&MSI for 3 separate runs (the three sets of image chips). As noted in Table 1 the NNMSE algorithm has significantly less false alarms than OptLin. Each set of image chips in Figure 8 is sorted in decreasing order using the neural net output. Note that for each run the higher valued chips are very similar between the runs; it is the lower value chips that differ. Also note that the NNMSE false alarms do not contain the large numbers of breaking waves chips that the OptLin algorithm contains; it appears that the neural net can weed many of these out.

4.0 Performance on Full Images

We now look at the algorithm performances on full images. This will test how well the detection algorithms operate when the encounter "new" background image chips that may not be represented in the training set. We have set up two sets of image tests. The first set contains 4 WorldView Ortho image tiles from the Valdes 2012 data set (these are R08C2, R08C3, R09C2, and R09C3) and 4 WorldView Ortho image tiles from the Valdes 2014 data set (these are R12C2, R12C3, R13C2, and R13C3). Each of these has some set of manually detected whale signatures which we will assume are the only whale signatures in the image tile. Thus every detection that is not one of these will be considered a false alarm. This means that for the detection algorithm results from the entire tile we can calculate NMD and NFA. The second set of images will be the WorldView Basic Maine images that we recently received. These are P001, P002, and P003 for both 15426304010 and 15426329010. These are the images that initially generated way too many detections in running the old detection codes. We assume that there are no whale signatures in these image (though see below for some possible ones) so all the detections are false alarms.

For each image, we record the number of times that we apply the detection filter. This will vary from image to image depending on the land within the scene (and if there is any zero-padding along the edges). We then normalize the NFA we generated for each image into our NFA/Tile metric by performing NFA*633536/(total number of filter applications). This allows us to compare NFA/Tile values across images that have different numbers of filter applications. But note that this will be the NFA/Tile for the given detector and threshold, not normalized to PD=0.90

Table 2 shows the results for the image tests. Each line of the table is for one image; the name of that image is in the first column. The second column labelled "NTARG" is the number of manually extracted whale signatures within that image only for the 8 Valdes images. The third column labelled "NFILTERS" is the number of times we applied the detection filter to that image. Note that the first 8 Valdes images are all WorldView Ortho tiles, so the maximum value that this number could be is 65536. The other columns are then the detector results for the two algorithms (OptLin are the next six columns and NNMSE are the last six) for the three different types of normalization. The NNMSE results use a threshold of 0.5 to keep false alarms low. For

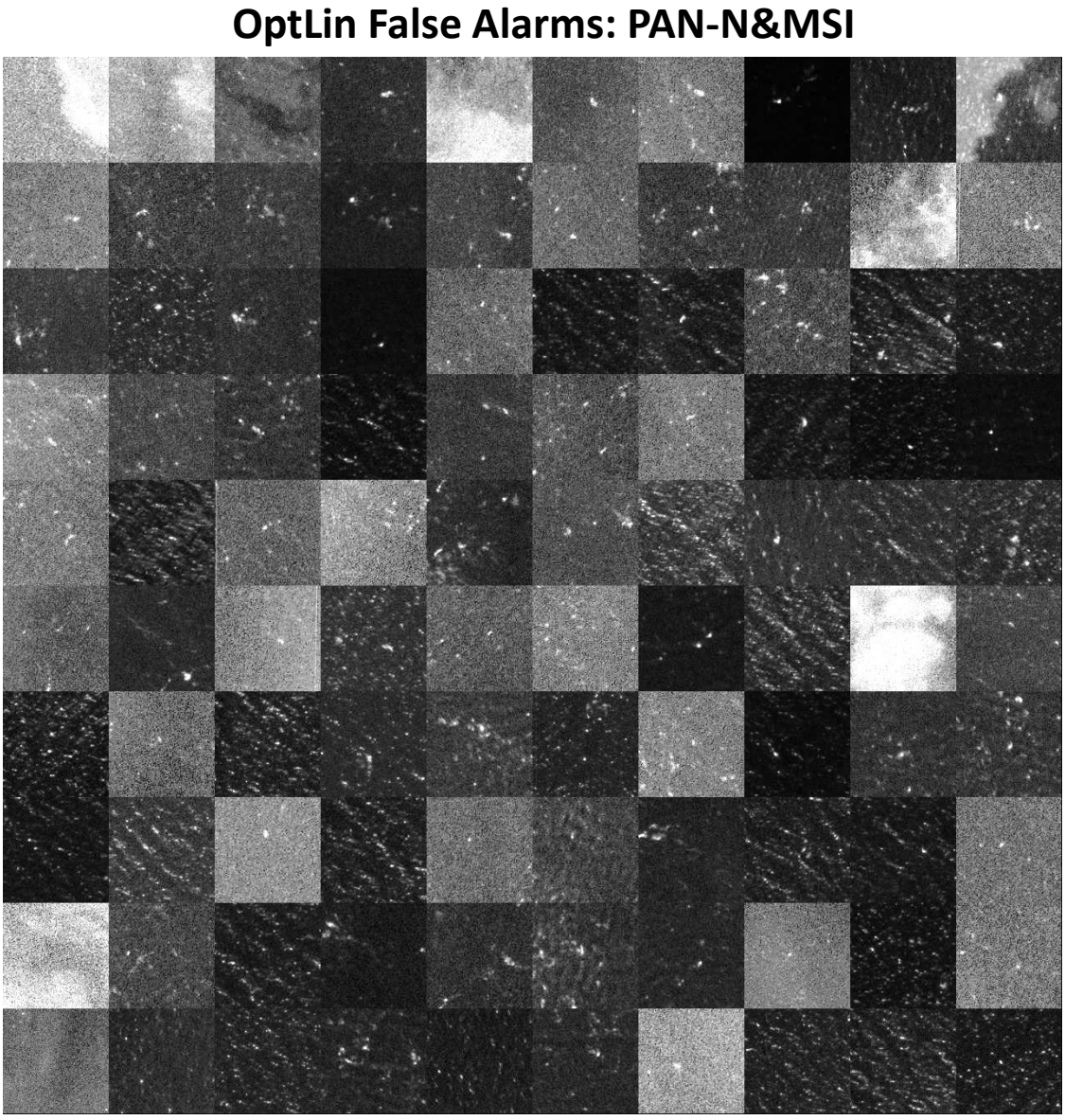

Figure 6: First 100 false alarms from the training set using the OptLin detection algorithm with the PAN-N&MSI normalization option. These are dominated by breaking wave signatures.

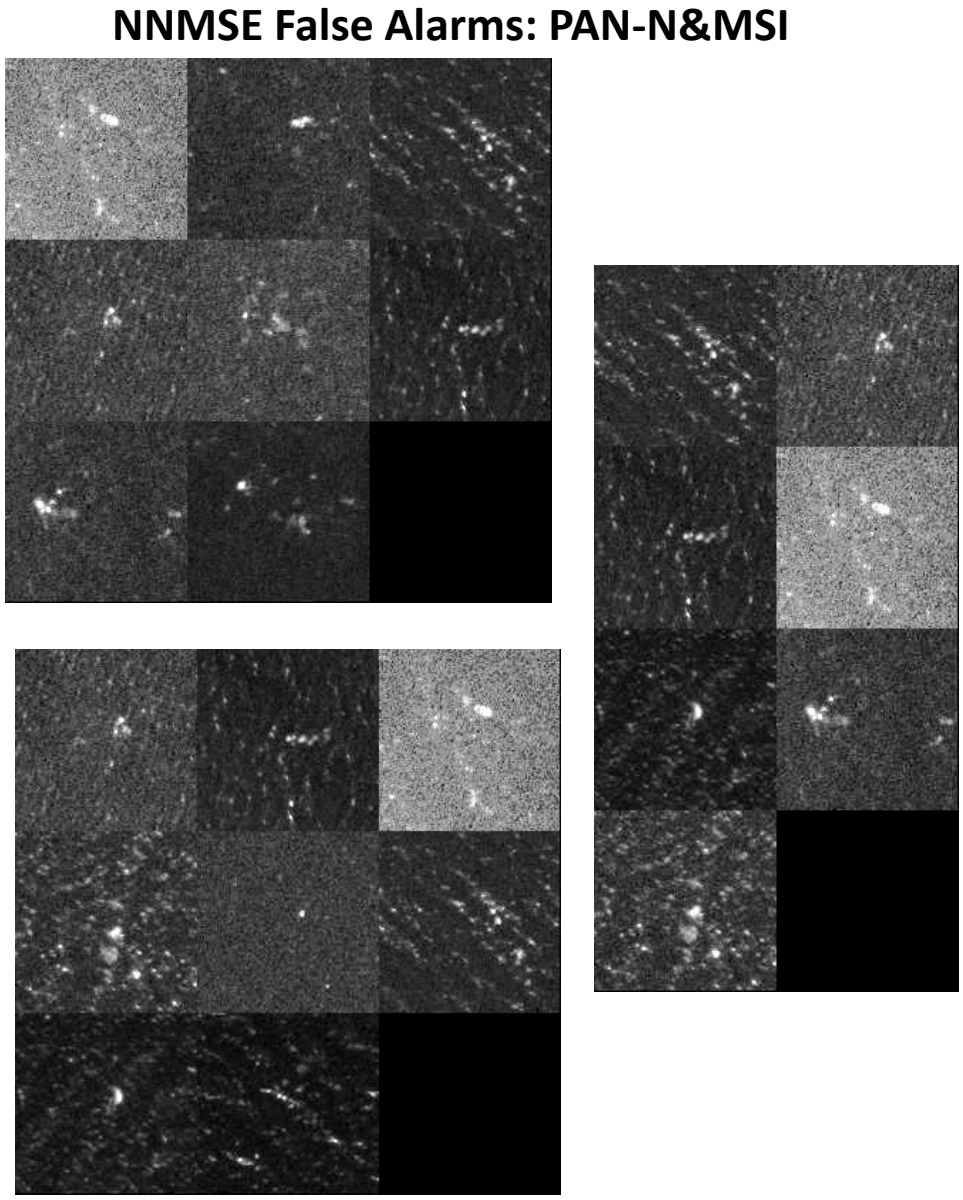

Figure 7: All the false alarms from the training set using the NNMSE detection algorithm with the PAN-N&MSI normalization option. Each set of image chips is the full number of false alarms from each trial run; much less false alarms than with OptLin.

			OptLin						NNMSE					
			PAN-N & MSI-N		PAN-N & MSI		PAN & MSI		PAN-N & MSI-N		PAN-N & MSI		PAN & MSI	
Image	NTARG	NFILTERS	PD	NFA/Tile	PD	NFA/Tile	PD	NFA/Tile	PD	NFA/Tile	PD	NFA/Tile	PD	NFA/Tile
R08C2	44	48656	0.93	89	0.93	54	0.93	100	0.89	7	0.93	104	0.91	12
R08C3	32	22458	0.88	292	0.88	125	0.88	222	0.88	12	0.94	499	0.88	137
R09C2	7	64516	1.00	2	1.00	1	1.00	10	1.00	2	1.00	6	1.00	5
R09C3	8	42418	0.88	11	0.88	9	0.88	23	0.88	11	0.88	12	0.88	22
R12C2	1	34739	1.00	2703	1.00	4562	1.00	12832	1.00	347	1.00	308	1.00	302
R12C3	33	54610	0.97	71	1.00	253	1.00	1295	1.00	8	1.00	17	1.00	23
R13C2	15	64516	0.87	355	0.87	650	1.00	3604	0.87	51	0.80	42	0.80	45
R13C3	3	55880	1.00	13	1.00	19	1.00	174	1.00	7	1.00	4	1.00	2
4010P1		2721		2168		1855		2987		771		289		265
4010P2		40187		6998		6342		9049		2942		475		323
4010P3		104880		14005		13425		15269		6613		844		410
9010P1		178968		580		391		664		0		0		0
9101P2		284749		724		956		1272		120		46		58
9010P3		298350		2185		3363		3436		10		0		0
Valdes Avg			0.94	442	0.94	709	0.96	2282	0.94	56	0.94	124	0.93	68
Maine Avg				4443		4389		5446		1743		276		176

Table 2: Detector results for the full image tests.

each algorithms/normalization we show PD for the Valdes images and NFA/Tile (normalized as discussed above) for all the images. The last two lines show the averages of these values for the first 8 Valdes images and the last 6 Maine images. The NNMSE results in Table 2 are just from one trial run (the first one we did to generate a neural net); we did not do multiple trials and average them for these results.

We can compare the average results in Table 2 to the results from the training set in Table 1. To normalize the NFA values in Table 1 we use that fact that there are 9340 background vectors so we have that NFA/Tile=NFA*65536/9340. Applying this scale factor to the NFA values generates NFA/Tile for Optlin of 1207-1691 with an average of 1509, and for NNMSE (using the threshold of 0.5) of 67-83 using un-normalized MSI and 251 using normalized MSI. Comparing these to the values in Table 2: the OptLin Valdes average ranges between 442-2282 with an average of 1144, NNMSE for un-normalized MSI ranges in 68-124 with an average of 96, NNMSE for normalized MSI is 56. Comparing averages, for OptLin it is 1509 for the training set and 1144 for the Valdes images. For NNMSE it is 75 for the training set (MSI un-normalized) and 96 for the Valdes imagery. So both of these are very consistent. However for normalized MSI we have a larger difference; 251 from the training set and 56 from the Valdes imagery. Looking at PD, the training set results noted above ranged from 0.96 to 0.98. The results in Table 2 range from 0.94 to 0.96. So very consistent, though with these images we do not achieve the higher 0.98 PD for the OptLin algorithm.

The Maine images NFA/Tile are generally much higher than the Valdes images. Visually the Maine images have a lot more ocean clutter, including wave signatures and breaking wave signatures. These are generating more false alarms. For OptLin and NNMSE with normalized MSI the Maine NFA/Tile values are much higher. However, note that for NNMSE using un-normalized MSI they are higher (176 and 276) but not by that much compared to the other results. It does appear that NNMSE with un-normalized MSI is more robust against these "new" false alarm signatures.

Which normalization is best? For the NNMSE results in Table 2, all three normalizations are essentially equivalent for the Valdes averages, but for the Maine imagery PAN&MSI generates less false alarms, though PAN-N&MSI is close. So similar to the training set results, for NNMSE PAN-N&MSI or PAN&MSI are the best. For OptLin, PAN-N&MSI-N gives the lowest NFA, but it is only slightly better than PAN-N&MSI. PAN-N&MSI-N gives the best PD for the Valdes imagery.

Which algorithm is best? For the Valdes imagery, PD is essentially the same for both OptLin and NNMSE. Across all 14 images, NFA/Tile is significantly less for NNMSE than for OptLin. As with the training set results in Table 1, NNMSE performs better than OptLin.

Figure 8 shows the missed detection from running the algorithms on all 8 Valdes test images for OptLin (left image) and NNMSE (right image) using the PAN-N&MSI normalization. Note that they are very similar. However comparing to Figures 4 and 5 we can see that the set of missed detections when we run on the full imagery is different than when we run on the training set. This is most probably due to the fact that when we run on the entire image we may not be centering on the same bright pixel when we generate the feature vector that we do when we run the training set. Based on the results in Figure 8 this appears to be true for both OptLin and NNMSE (since they are generating very similar sets of missed detections). As shown in Table 2 there were 143 whale signatures in the 8 Valdes test image and we found all of them except the ones shown in Figure 8.

The results from the training set (Table 1) and the full images (Table 2) consistently indicate that the best performance comes from using the NNMSE algorithm with either PAN-N&MSI or PAN&MSI. In both cases it seems a wash whether we normalize PAN or not, but we definitely should not normalize MSI.

Finally, note that with the new improvements we can now reasonably review full images for whale signatures. For all 6 Maine images the total number of detections using NNMSE with PAN-N&MSI was 1853, versus ~40,000 using the old approaches. We performed a manual review of all the detections, and Figure 9 shows the ones that looked interesting. These are all pan-sharpened images that use both the PAN and MSI image data. The top set of images are these white regions in the water; perhaps floating vegetation, or breaking water, or whales? The bottom set are somewhat more traditional whale-like signatures.

5.0 Additional Capabilities

Automated Generation of Water Mask

We have a new capability that will automatically generate a mask for any TIF image that indicates water regions and land regions. It uses the TOPEX worldwide bathymetry data base to generate a coarse resolution lane/water map where water is defined as any region where the TOPEX depths are > 10m. Fine-tuning of the mask is then done by finding the median image value for water and for land, and generating a threshold that is half-way between these. This threshold is then applied to each pixel in the image that has a TOPEX depth > 10m to set it to either water (less than the threshold) or land (greater than the threshold).

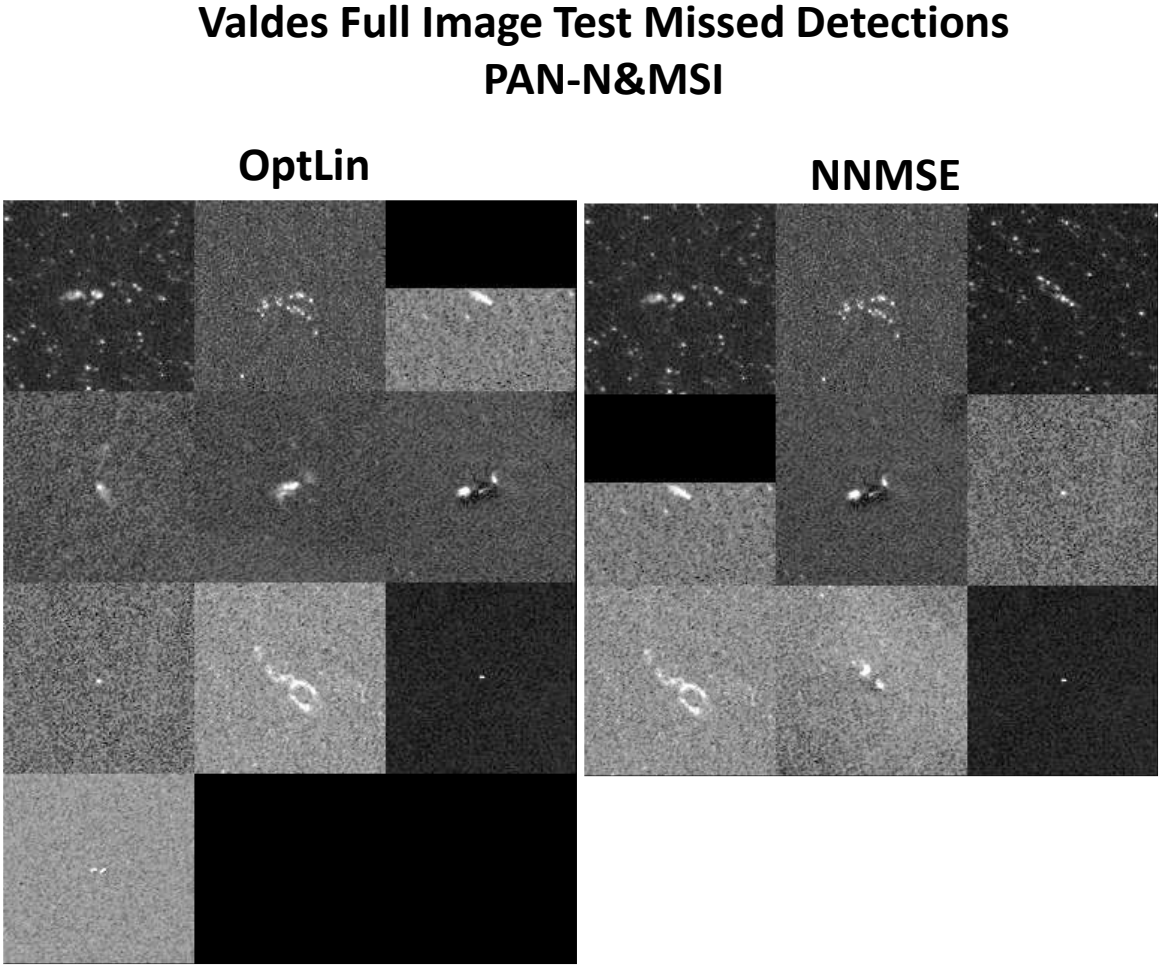

Figure 8: Missed detections from the 8 Valdes test images using the OptLin detection algorithm (left) and the NNMSE detection algorithm (right) with a threshold of 0.5. The PAN-N&MSI normalization option was used for these results.

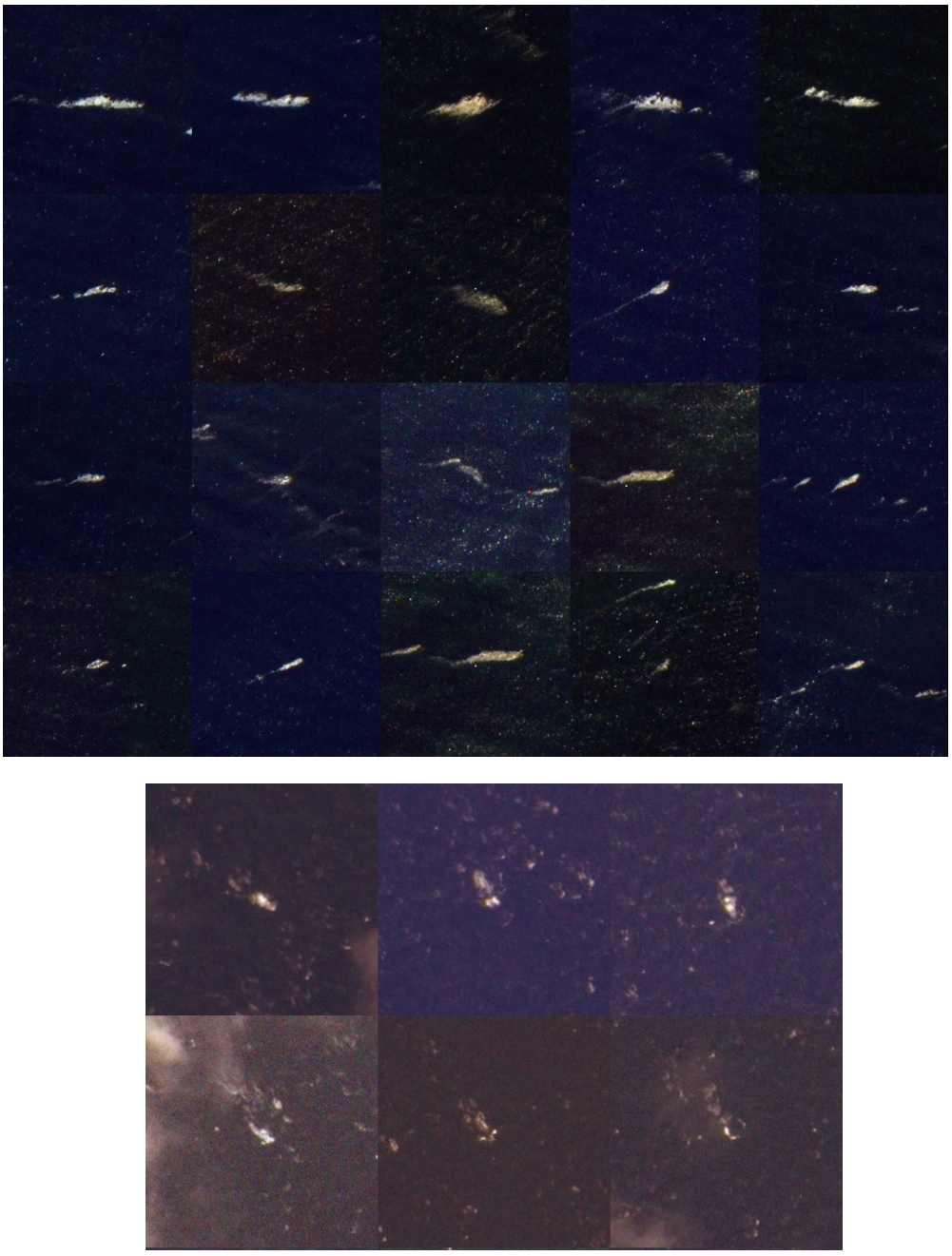

Figure 9: Interesting detections from the 6 Maine test images.

Figure 10 shows three examples taken from three of the new Maine images. The left column is the original PAN TIF image, the middle column is the resulting water mask (red means water, blue means land, I know….it is the opposite of what it should be!), and the right column shows both of them together. One can see that the coarse TOPEX bathymetry means that our limit of 10m depths is not allowing us to capture the water close to shore, but this is probably OK for the whale detection problem. Note that in deeper water, the threshold fine-tuning is doing a good job of isolating the small islands, rocks, etc. as land.

Overall this is significantly decreased the amount of land detections we get when we run the detectors on full images. Previously we had to manually derive a water polygon which was time-consuming for each image. This new approach is done automatically without any user input.

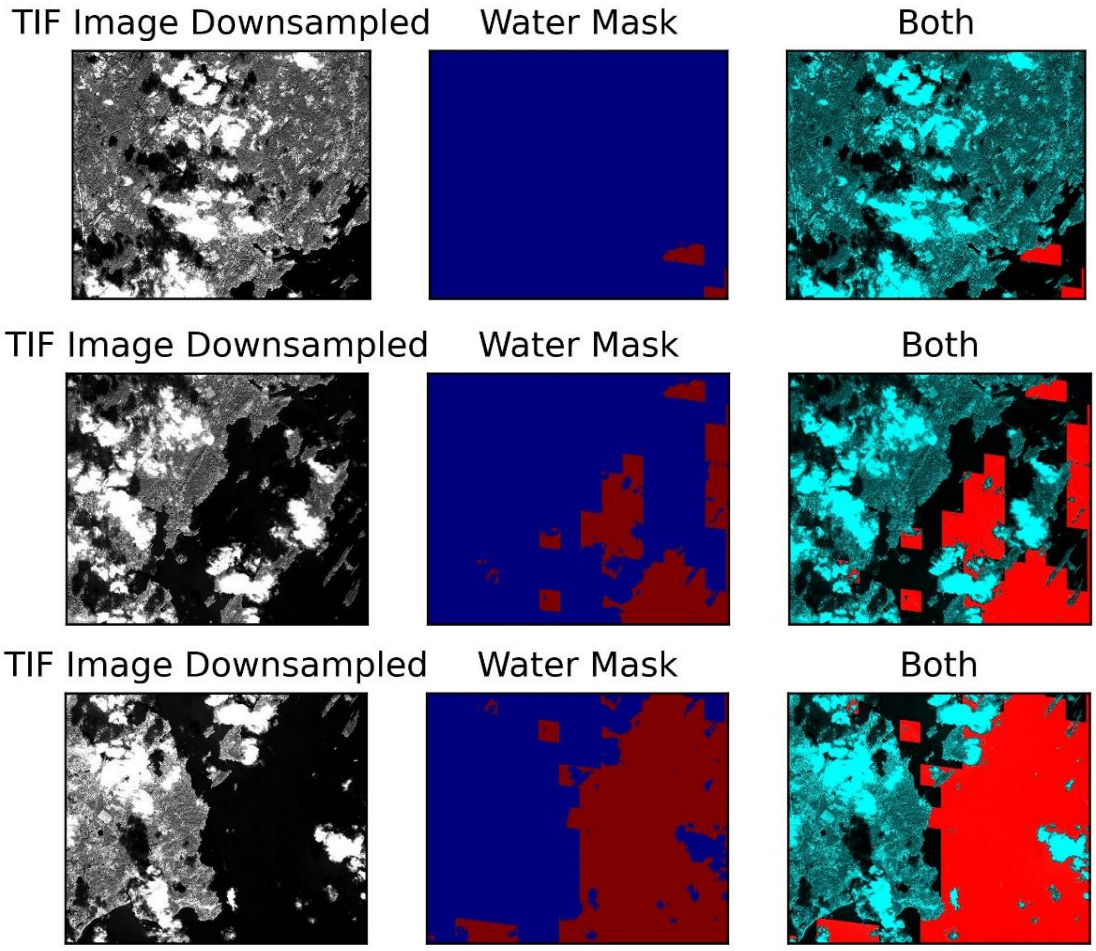

Figure 10: Example water masks from 3 Maine images. Left column shows the original PAN TIF image, middle column shows the water mask derived from a combination of TOPEX bathymetry and image thresholding, right mask shows both.

Principle Components Analysis of Feature Vectors

We have added a capability to perform a Principle Components Analysis (PCA) of the feature vectors. PCA is a standard technique for analyzing the inherent dimensionality of a dataset and helps to visualize what it looks like in N-dimensional space. Given a dataset of M vectors (each of length N), PCA is performed by first calculating the covariance matrix of the M vectors normalized by subtracting the mean and dividing by the standard deviation of each N entry. We then find the eigenvectors and eigenvalues for the covariance matrix. The eigenvalues give us an estimate of the dimensionality of the dataset; the number of normalized eigenvalues (normalized by dividing each eigenvalue with the sum of all eigenvalues) we need to sum to get close to 1.0 represents the number of independent values in the vectors. The eigenvectors represent the directions in N-dimensional space we need to project the data to get down to the smallest number of features to still represent the full dataset. We can then look at the top 3 eigenvetor projections to get a sense of what the N-dimensional dataset looks like.

We developed a tool that generates eigenvectors and eigenvalues for: (1) both background and target feature vectors considered as a single dataset; (2) just the target feature vectors; and (3) just the background feature vectors. Figure 11 shows the results when applied to the feature vectors generated using PAN-N&MSI. The top row (after rotating the figure 90 degrees CCW) shows the results applying PCA to the entire set of feature vectors (both target and background); the middle row applies PCA to only the target vectors; the bottom row to only the background vectors. The first column plots the eigenvalues. What is interesting about these is that the fall-off is rather slow. Normally the eigenvalues become too small to worry about around 2-4 from the maximum. But these results show that the dimensionality of the feature vectors is around 11. Since we have 23 features, this means that about half of the data is independent; which is a lot. The second – fourth columns show plots of the top 3 eigenvectors (top mean having the largest eigenvalues) for each PCA. These represent the directions in N-space that represent the largest variation in the dataset. Because of the slow fall-off of eigenvalues, these three only represent about 71% of the total dataset variance. What is interesting though is to look at the first eigenvector which represents the direction of largest variation. Note that the most important components (in the sense of having the highest absolute value) are all the normalized PAN statistics with levels of around 0.3. Next, with levels around 0.2, are all of the anomaly metrics and the MSI band means for all bands except for the two near-IR bands. Thus all of these features appear to be important for characterize the full extent of the feature vectors. Note that for all three analysis (using all data, using only targets, using only background) these general trends are still true, but the details between the target (i.e. whales) and background results are interesting. For the whale data, the Coastal, Blue, Green, and Yellow bands have much higher weights than they do for the background data whereas the image metrics have much lower weights. This indicates that the whale data is represented better with the lower-band means, whereas the background is represented with the image features.

Better Review of Detections

Because we can now automatically attach MSI image pixels to PAN image pixels, we can generate pan-sharpened imagery to the user for reviewing the detections. This makes it somewhat easier to differentiate between ships/boats and possible whale signatures. It also

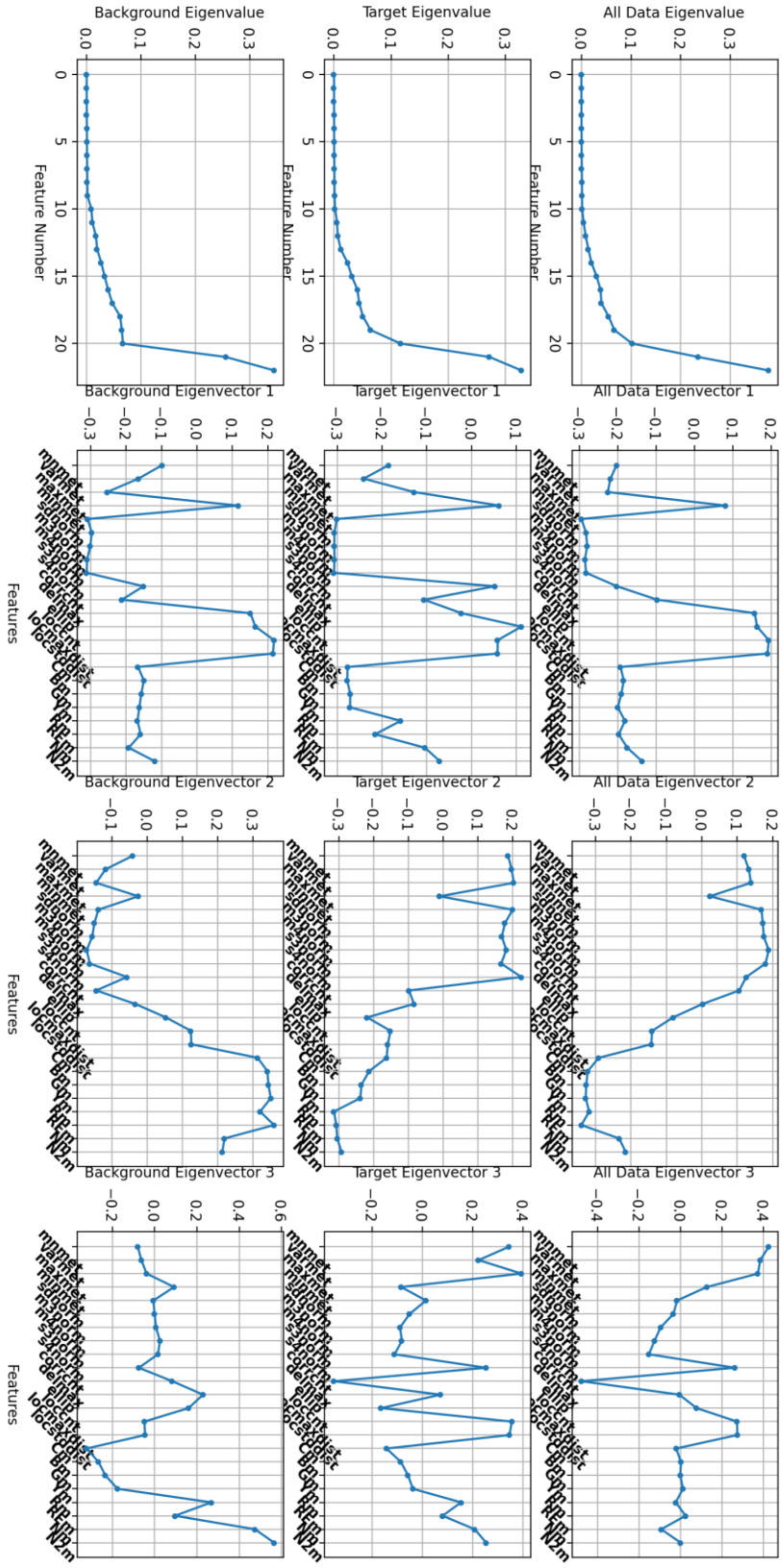

Figure 11: PCA analysis on the PAN-N&MSI feature vectors (see text for discussion)

makes land feature more apparent. We have coded three pan-sharpened algorithms: Bovey, ESRI, and Simple Mean. In addition, we have added the ability to re-order the detections based on the detection metric (either the output of the neural net for NNMSE or the classification project value for OptLin) so that we present to the user the detection image chips with the highest detection metrics first, and thus with the highest probability of being a whale signature.

Figure 12 shows an example. These are the first set of detections that were generated using the NNMSE PAN-N&MSI algorithm on the Valdes 2012 Ortho Worldview image R08C3. There were 30 manually-derived whale signatures detected in this image and they are all shown in the top four lines of Figure 14 (i.e. they had the highest detection metrics). All the rest of the detections are false alarms from background chips. Note that for this image some land has still crept into the water mask, but the pan-sharpened images make the land chips obvious.

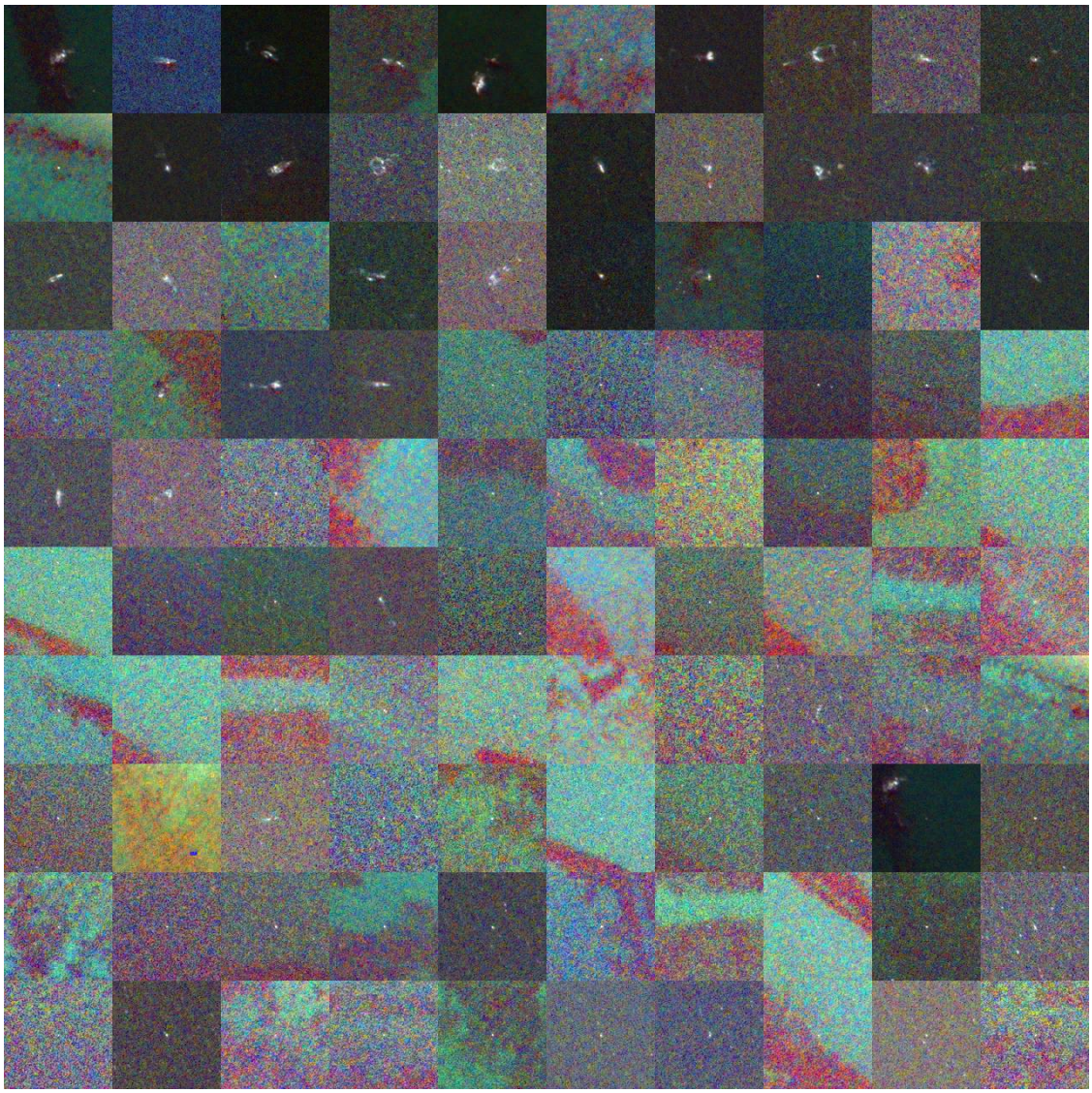

Figure 12: Example output image for manual review of detections that uses pan-sharped imagery and orders the chips from high to low detector metric values.

REFERENCES:

[1] Wackerman, C., “Initial Results in Analysis of Whale Signatures in Multi-Spectral and Panchromatic Imagery”, NRL Technical Report, 09 October 2020.

[2] Wackerman, C., “Initial Results in Analysis of Whale Signatures in Multi-Spectral and Panchromatic Imagery: Revision 1”, NRL Technical Report, 05 November 2020.

[3] Wackerman, C., “A Semi-Automated Anomaly Detector for Whale Signatures in Multi-Spectral Imagery”, NRL Technical Report, 13 May 2021.

[4] Wackerman, C., “Improved Feature-Based Detection of Whale Signatures in PAN Imagery and Comparison Between Optimal Linear and Neural Net Algorithms,” NRL Technical Report, 02 September 2021.

[5] Wackerman, C, ”Automated Detection of Whale Signatures in PAN and MSI Imagery Using Features Vectors Combined With Neural Net Algorithms,” NRL Technical Report, 05 July 2022.

Appendix A:

Table A1 shows the detection metrics for each of the trial ones using the training set. The average of these values over each normalization option is what was put into Table 1 in the text.

Algorithm	Features	Metrics							Thres=0.5	
		Fischer	MSE	ROCC Area	ROCC Dist	NFA/Tile PD 90	NMD	NFA	NMD	NFA
NNMSE	PAN-N & MSI-N	9.4	0.0079	0.993	0.962	344	16	242	62	34
	PAN-N & MSI-N	9.5	0.0079	0.994	0.961	323	17	229	62	40
	PAN-N & MSI-N	9.6	0.0078	0.995	0.996	302	23	170	62	34
	PAN-N & MSI-N	9.8	0.0078	0.995	0.964	309	17	192	62	36
	PAN-N & MSI-N	9.1	0.0081	0.993	0.957	351	21	213	63	30
	PAN-N & MSI-N	11.2	0.0083	0.995	0.969	358	15	154	61	41
	PAN-N & MSI	43.3	0.0028	1.000	0.992	14	1	79	11	16
	PAN-N & MSI	26.3	0.0031	1.000	0.990	7	5	42	26	9
	PAN-N & MSI	26.2	0.0031	1.000	0.990	7	2	89	27	8
	PAN-N & MSI	24	0.0030	1.000	0.990	7	3	80	27	7
	PAN-N & MSI	26.9	0.0033	0.999	0.992	21	2	66	29	8
	Pan & MSI	30.7	0.0029	1.000	0.990	0	1	90	21	13
	Pan & MSI	25.3	0.0030	1.000	0.988	14	4	90	23	10
	Pan & MSI	27.8	0.0032	1.000	0.991	0	4	52	21	13
	Pan & MSI	29.6	0.0029	0.999	0.992	7	2	69	20	11
	Pan & MSI	28.1	0.0028	0.999	0.990	28	3	78	17	12