Note: **This technical paper is submitted for journal review. The id of this technical paper is TR2018/3.**

Machine Learning Applications in Detecting Rip Currents from Images

Corey Maryan [2, 3], Md Tamjidul Hoque[1, 2,*], Mahdi Abdelguerfi[1, 2], Christopher Michael[3], Elias Ioup[3]

Email: {cmaryan, thoque, mahdi}@uno.edu, {chris.michael, Elias.Ioup}@nrlssc.navy.mil

[1]Canizaro/Livingston Gulf States Center for Environmental Informatics, University of New Orleans, New Orleans, LA 70148, USA,

[2]Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA,

[3]US Naval Research Laboratory, Stennis Space Center, MS 39529, USA.

*Correspondence: thoque@uno.edu; Tel.: +1-504-280-2406

Abstract

Annotated rip current images are used in oceanographic climate studies and they are mostly annotated by hand. For thousands of images, this manual annotation becomes difficult. In recent years, object detection has become a successful approach for identifying regions of an image. There are several different algorithms currently used for detecting objects from images, however, there is no software tool to automate the detection of rip current from images. While developing the tool, this paper describes a comparison between the Viola-Jones object detection algorithm, deep learning or, convolution neural networks, and a meta-learner on a dataset of rip current images to find the most suitable algorithm for detecting rip currents. These algorithms are run on a benchmark of rip current images and compared based on the detection rate and the number of false positives contained in each benchmark image. In addition, this paper makes two more contributions. One is a new set of Haar features based upon the originals. The other contribution is the building up of a meta-classifier, which combines the outputs of several machine learning techniques to increase accuracy of the rip current classifier by reducing the false positives of the state-of-the-art approach.

Keywords: rip current; object detection; deep learning; support vector machine; principal component analysis; meta-learner.

1. Introduction

A rip current is a seaward force, is perpendicular to the shoreline, and is caused by two waves breaking in an along-shore direction [1]. Images of these rip currents are typically involved in oceanographic climate studies [2], however, must be manually annotated by hand as there has been little research done in automatically detecting rip currents from images. There exists a backlog of thousands of shoreline images taken daily [3], nevertheless, the manpower to search through each image does not. One possible solution to this problem is to combine machine learning techniques and object detection.

Object detection has become a popular approach to find certain regions within an image [4][5][6]. Object detection methods are applied to a wide variety of objects, which makes applying such techniques to rip current images intuitive. There are several popular detection algorithms to choose from in the literature, yet, there is no research supporting which algorithm is appropriate for rip current detection. In order for a proper method to be found, a comparison of current, state-of-the-art models is done to determine the most worthwhile classifier. The methods chosen for comparison are the Viola-Jones object detector, deep learning particularly convolutional neural networks (CNN), support vector machines (SVM), max distance from the average rip current image using principal component analysis (PCA), and the meta-classifier [7].

The TensorFlow [8] framework builds and runs the convolutional neural networks, the OpenCV [9] package runs Viola-Jones, and the Scikit-learn [10] package for Python runs the basic classifiers, which build the meta-classifier. The Scikit-learn package also runs the SVM. The max distance from the average rip current image is generated and applied for classification through principal component analysis (PCA) in Matlab.

This paper provides a comparison of current state-of-the-art object detection models, which assists researchers in automated rip current detection from images and determines the most appropriate methods for identifying rip currents. The goals of the research are to (i) identify the most suitable rip current detector to assist researchers in rip current detection; (ii) find features suited for rip current images; and (iii) improve upon the state-of-the-art detection models when detecting rip currents. The results of this study analyze each model with reasonable performance metrics from previous research.

Another contribution of this paper is a new set of Haar features, which are optimized for rip currents. These are based upon the features discussed in the Viola-Jones algorithm [4], are created by changing different regions extracted from a 3 by 3 matrix, and correspond to the average image of a rip current.

Finally, a meta-classifier is presented as a third contribution. This classifier is built with the class confidence values of several different machine learning algorithms. These algorithms train on new Haar features introduced by this paper. These confidence values help make up for what a model may lack, which increases accuracy [7].

For the rest of the paper, section 2 describes the background associated with the detectors. Section 3 identifies how each model is built and identifies the dataset for each model. Section 4 describes the results of the experiments. Section 5 contains the conclusions.

2. Background

This section contains a background of the rip currents and object detection research fields, which are directly related to the research.

2.1 Rip Currents

Rip currents are an area of nearshore research for oceanography. Climate studies use annotated rip current images, but they are manually annotated by hand [2]. The time required for annotation makes conducting this type of research challenging when images amount in the thousands. There have been attempts at detecting rip currents with semi-automated methods [11, 12], however, no fully-automated methods have been explored. The first semi-automated method locates maximum and minimum intensity values to identify rip currents. A human is required to correct this algorithm after it makes its initial predictions. The other algorithm needs a human to manually digitize each rip current so the algorithm can make a prediction on the data after the images are corrected for noise. These methods still need a similar amount of time because of the eventual human involvement. Creating a fully-automated method has possible applications for machine learning and object detection.

2.2 Machine Learning based Object Detection

Machine learning employs algorithms that fit data to a model [13]. These models can then make predictions on data samples that the model has not come across. Fitting data to make predictions is the basic process of learning. To learn, models train on a set of features, which describe each sample in a numerical fashion. A model making prediction on regions of an image after training on features extracted from the image is known as object detection.

In recent years, object detection has become a popular method to automatically identify regions of an image [5, 6, 14]. Object detection has numerous algorithms to choose from because of the wide variety of machine learning techniques in literature. Therefore, a comparison of such algorithms is valuable when deciding which one is most appropriate rip current detection. Rip currents are a lack of focus for object detection, which also creates a critical need for adequate rip current features in addition to a comparative study of algorithms. The optimized features increase the accuracy metrics by which the models are compared. Comparing models requires relevant, realistic metrics. In object detection, a couple of the most commonly applied metrics are detection rate and false positive rate. These metric represent how accurately a model can identify rip currents and avoid misclassifying non-object samples, respectively. Examples of popular object detection algorithms are Viola-Jones [4] and convolutional neural networks [5, 6, 14].

Viola-Jones has success in the area object detection, specifically with faces [4]. This is due to its speed and robust nature. Viola-Jones employs the Ada-boost algorithm, which is highly resistant to overfitting [15]. This is a desirable quality as the dataset contains a high amount of variance. Viola-Jones contains a series of layers, which detect the object in question. Every layer has its own detection rate and false positive rate. The total rates for the cascade is a product of each layers' rates. Viola-Jones has a set of optimized features for face detection, called Haar features. These are rectangular regions that correspond to different areas of the face. They are easily applicable because of their instance evaluation speed.

Convolutional neural networks also have success in detecting any object [5, 6, 14]. These networks create their own features for detecting an object, which are based on image filters. Over thousands of sample, the networks learn which filters are important to a specific object. The networks then apply the learned filter to detect the object sought after. These networks need a large amount of samples to train on because they over fit easily on a small dataset due to being formulated as high order polynomials for flexibility.

3. Detection-Model Comparison

This section describes how each detector is setup and run for comparison, including: max distance from the average rip current image, SVM, convolutional neural networks, Viola-Jones, and the meta-classifier. The detectors are compared on a benchmark of rip current images with detection rate, false positive rate, false positive count, and accuracy as metrics.

3.1 Dataset

The dataset contains 514 rip current examples, including the benchmark [16]. These rip currents are 24 by 24 images taken from a backlog of larger beach images [3]. Images average about 2 to 4 rip currents per image. In total, there is a little over 100 large images of shorelines that contain the extracted rip current samples. Small samples are taken by hand. The large images of the shoreline are from the cameras located at Duck, North Carolina and Secret Harbour, Australia. These two sites contain the highest quality rip current samples. An example image with rip currents from these sites is seen in Figure 1. In Figure 1, the shoreline is seen from a bird's eye view. This image contains 5 annotated rip currents indicated by red rectangles. These types of images are given, as input, to the detectors. The rip current samples, extracted from these large images, are all normalized to 24 by 24 for efficient Haar feature calculation. The training set contains 461 rip current images while the benchmark has 53 rip currents in 12 large images of shorelines. An evaluation dataset is typically made from 10% of the total dataset, hence the 53 rip currents.

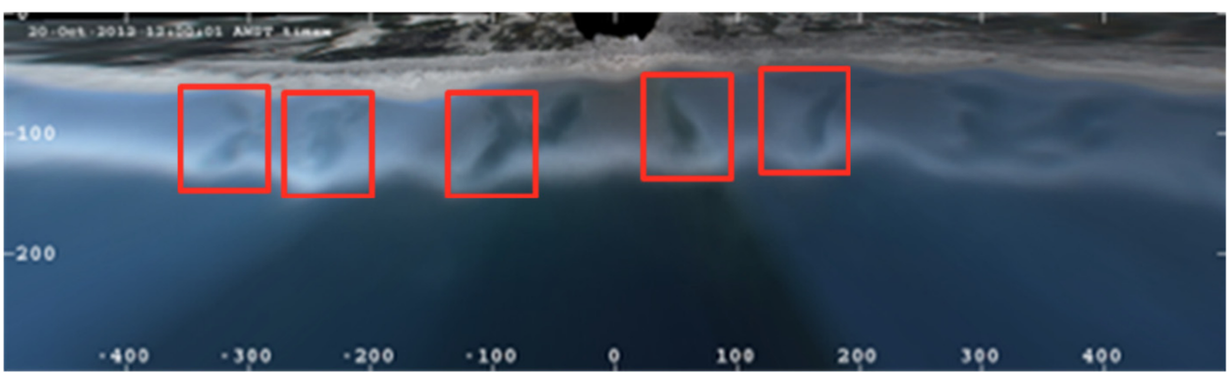


Figure 1. An example image of the Secret Harbour, Australia shoreline.

OpenCV has a tool for creating new, positive samples by applying distortions to every sample. These samples are then imposed onto a larger background image. A warp factor of 0.1 creates a dataset of 4000 rip current images [17]. Convolutional neural networks need to train on many thousands of images to attain decent results [14]. Therefore, creating a larger dataset is necessary. This dataset is compared with the small dataset since the OpenCV library can accept either as input. A comparison of datasets helps determine the most effective type of model.

3.2 Threshold from the Average Rip Current Image using PCA

The process of object detection is simplified if a range of values representing rip currents is found. Normally, a model must train on a dataset of positive and negative samples. Eliminating the negative samples simplifies the dataset. To this end, principal component analysis (PCA) is employed [18]. This method reduces the number of dimensions in the 24 by 24 image dataset of a rip currents into a number of chosen components from the Eigen vector. Each component is projected toward to generate a max distance from the average rip current image. Around 300 max distances from the average are generated from the components. A maximum distance from the average feature vector is found by first finding the average feature vector. Projecting a rip current vector toward 1 component generates a rip current feature vector of size 1. Finding the greatest distance between any feature vector and the average feature vector creates the max distance from the average. This is applied as a threshold for rip current identification. If a new data sample projected toward a component generates a feature vector with a distance less than the max distance, then it is classified as a rip current. Otherwise, it is classified as not a rip current. The thresholds are tested for their detection rate and false positive rate on a test image set.

3.3 Support Vector Machines

Support vector machines are a popular classifier because of their accuracy with high dimensional data [19]. Images naturally contain a large number of dimensions. Consequently, they are compared on the rip current dataset with the other detectors.

(A) **(B)**

Figure 2. Image **(A)** is an examples of a normal rip current image in gray scale. Image **(B)** is a segmented version of the same image.

The SVM features are averages of the Haar features from the Viola-Jones algorithm. When the original features are combined with the new set, the feature set contains 10 types of features with a total of over 200,000. This is a descriptive set with success in object detection, but over 200,000 is too many dimensions to learn in a reasonable amount of time and memory usage. Instead, an average of each type of Haar feature is taken at the cost of descriptive information. In addition to these averages, circularity [20] and black-white ratio are added to the feature vector for each sample. Circularity is taken from the segmented image. The segmented image is generated from a binary image with Matlab routine *bwconncomp*. This method finds all connected components in the binary image. An example is shown in Figure 2. The rip current object is in black while the background is in white. The equation for circularity is shown in (1).

$$\text{Object's Circularity} = 4\pi(Area/Perimeter^2) \tag{1}$$

Here, for a segmented object, *Area* is the total count of how many black pixels do not have white neighbors, while *Perimeter* is the total count of how many black pixels have white neighbors, where neighboring pixels are defined, in 2D, as the 4 non-diagonal but topologically adjacent pixels. This circularity-measurement (1), assumes that the object is in black and the background is in white.

Black-white ratio is taken from an image of a rip current where each pixel is converted to totally black or totally white, depending on a threshold. The black white images are generated from Matlab routine *imbinarize*. This routine uses a globally defined threshold to set all pixel intensities in the image to either 0 or 1. Black white ratio is chosen as a feature based on the intuition that similar, normalized objects will have a similar black-white ratio.

The Scikit-learn package for Python runs the SVM [10]. The SVM has a RBF kernel. Grid search optimizes the parameters, *C* and *gamma*, for the kernel. The results of the grid search for the small dataset of rip currents are $C = 4.0$ and $gamma = 0.00390625$. A robust scaler object [21] from the Scikit-learn package is applied to the rip current data, which scales data based on an interquartile range (IQR) between the 25th and 75th quartile. A range is created for each feature in the training set. The results for the SVM are based on 10-fold cross validation.

3.4 Convolutional Neural Networks

Recently, convolutional neural networks have become a popular method for deep learning and object detection [5, 6, 14]. The convolutional neural networks in this comparison are built from the TensorFlow framework [8], which is developed by Google. TensorFlow's default configurations builds a variety of CNNs. Pre-built configurations and the framework make it easier to generate results since parameters are pre-defined. The CNNs that are chosen are the "Mobilenet" and "Inception" models. The Mobilenet model is geared more toward speed [5] while Inception is geared more toward accuracy [6]. These are chosen to evaluate a wide range of CNN capabilities. These networks train for 5 weeks on the OpenCV annotated dataset as they require thousands of images. The models are then run on the benchmark of rip current images and are compared alongside the other models.

3.5 Viola-Jones Method in OpenCV

The OpenCV package [9] has an implementation of the Viola-Jones algorithm for use with any object. A combination of different images create a set of cascades. There are 5 total cascades created. The first cascade trains on the small dataset of rip currents and negative images [16]. The second cascade is built with the small rip current dataset and a large dataset of surf zone negatives. The third cascade is built from small rip current images and a large negative dataset of any image. The fourth cascade is built with created negatives of the surf zone and the large positive image dataset. The last cascade is built with created negatives of any image and the large created dataset of rip currents [17]. The cascades are run on the benchmark of images after they train to completion. A false positive rate of 0.7 per layer and detection rate of 0.994 per layer train each cascade.

3.6 Meta-Classifier

The following section describes features that the meta-classifier trains on and how it is implemented.

3.6.1 Novel Features

The Viola-Jones Haar features are successful in detecting faces. Naturally, an optimized set of Haar features is needed to effectively detect rip currents. 19 new Haar features are created with a 3 by 3 matrix by changing the formula for calculating the difference of regions. The matrix for creating the Haar features is seen in Figure 3.

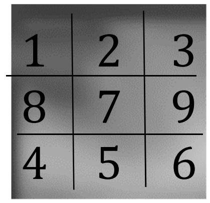


Figure 3. The Matrix of the new features laid on top of a rip current image. Each number represents area of space that can be extracted from the integral image to use in a Haar feature formula.

Table 1. The results of using each feature to train 10 layer cascades of weak classifiers. The Formula column refers to the matrix described in Figure 3.

Feature Pattern	**Formula**	**Detection Rate**	**False Positive Rate**
X	[1 + 3 + 4 + 6] - [7]	0.996	0.5
T_1	[1 + 2 + 3 + 5] - [7]	0.996	0.5
Inverted T_1	[6 + 4 + 5 + 7] - [2]	0.995	0.6
Three Columns	[3 + 9 + 6 + 1 + 8 + 4 + 7 + 5] - [2]	0.996	0.5
Cross	[2 + 7+ 9 + 8] - [5]	0.995	0.8
I	[1 + 3 + 4 + 6] - [2 + 7 + 5]	1	1
T_2	[1+ 3] - [2 + 7 + 5]	1	1
Short T_3	[1+ 3] - [2 + 7]	1	1
Inverted T_2	[6 + 4] - [5 + 2 + 7]	1	1
V	[1 + 3] - [7]	1	1
^	[4 + 6] - [7]	1	1
$[_1$	[5 + 7 + 2] - [6 + 3]	1	1
$[_2$	[5 + 2 + 6 + 3] - [7]	1	1
$]_1$	[5 + 2 + 1 + 4] - [7]	1	1
>	[1 + 4] - [7]	1	1
<	[3 + 6] - [7]	1	1
$]_2$	[3 + 6 + 9] - [2 + 5]	1	1
$[_3$	[8 + 4 + 1] - [2 + 5]	1	1
L	[6] - [2 + 7 + 5]	1	1

Subscript in the feature-pattern indicates the variations of that particular pattern.

Here, in Figure 3, each number of the matrix corresponds to a region of intensity values in the image. The average rip current image has 2 important regions: the middle of the image and the top-center of the image. These are regions 2 and 7 in the matrix (see Figure 3), respectively. Regions 2 and 7 help create the most accurate Haar features for rip currents. These features are tested by creating layer 1 of a Viola-Jones cascade 10 times over with a random set of negative images for each layer test. The performance metrics are averaged over the 10 built layers. The results of running the tests on each Haar feature are seen in Table 1.

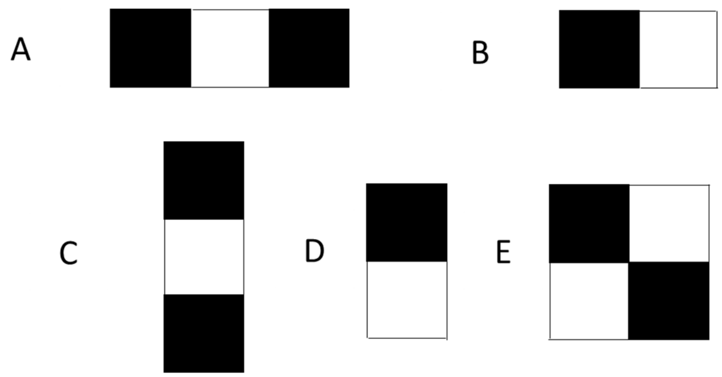


Figure 4. Example Haar features (A) The "Three Horizontal" feature, (B) The "Two Horizontal" feature, (C) The "Three Vertical" feature, (D) The "Two Vertical" feature, (E) The "Four" feature [4].

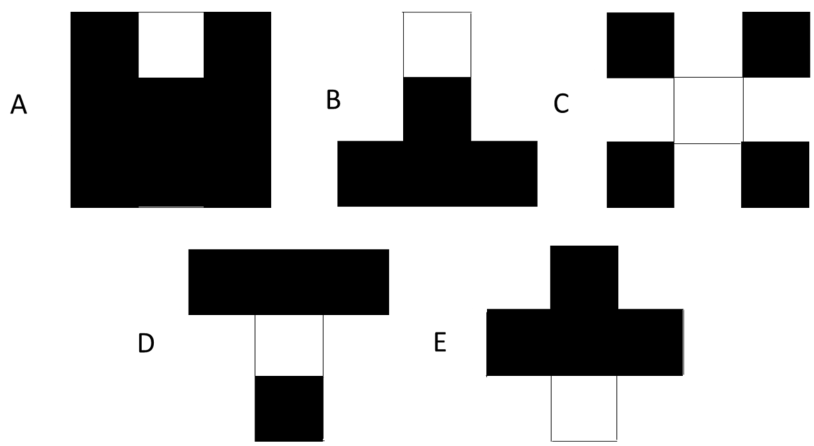


Figure 5. The Newly Added Features: (A) "Three columns", (B) "Inverted T_1", (C) "X", (D) "T_1" and (E) "Cross".

The formula for each feature in Table 1 is found by adding some combination of integral image regions from Figure 3 together then subtracting other regions from the total. For example, the formula for feature "X" is applied by first adding together the integral image for regions 1, 3, 4, and 6. Then, region 7 of the integral image is subtracted from it and hence, [1 + 3 + 4 + 6] – [7] (see Table 1, first row, 2nd column). In Table 1, "X", "T_1", "Inverted T_1", "Three columns", and "Cross" finish the test and are added to the total Haar feature space. Ada-Boost finds the most appropriate Haar features for rip currents from both the original (see Figure 4) and the new features (see Figure 5). Choosing appropriate features is done by first building a 10 layer Viola-Jones cascade. Then, each Haar feature result is appended to the feature vector for training until accuracy levels off for each basic model. The distribution for the final feature vector are 70% old 30% new.

3.6.2 Implementation of Meta-Classifier

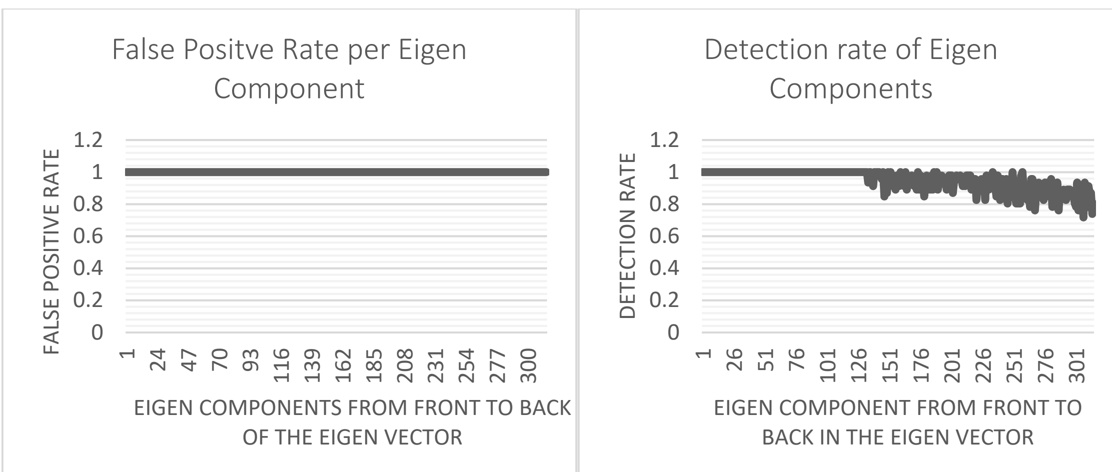
The meta-classifier trains on the confidence of previous models. Confidence values from other models alleviates what a model may lack [7]. Most of the Scikit-learn [10] models are used as basic classifiers to generate confidence values for each rip current sample. The basic classifiers include: SVM, neural network, decision tree, random forest, k-nearest neighbors, Naïve Bayes, bagging, and Ada-boost. Each model except for SVM run with the default Scikit-learn parameters. The parameters for SVM are described in section 3.3. The first 77 Haar features, chosen by Ada-boost, train each basic classifier. 10-fold cross validation attains the probability of the models. Every rip current sample has a training vector of 85 after adding the 8 confidence values from the basic models. The 85-feature vector trains the meta-classifier. Each model models in the Scikit-learn package that generate confidence values for training the meta-classifier are also are evaluated as the final meta-classifier model to find the best fit. The meta-classifier is added to the back of the Viola-Jones cascade to reclassify its output.

4. Results

This section describes the accuracy, false positive rate, and detection rate for each of the previously mention methods of comparison.

4.1 Threshold from the Average Rip Current Image using PCA

The detection rate and false positive rate for every max distance generated is shown in Figure 6. Not every component in the Eigen vector is displayed. A cut-off point is made once the components reach a value of 0. The detection rate tends to decrease as components with less variance are projected toward to generate the max distance from the average rip current image. The detection rate is decreasing because the max distance from the average is decreasing. Consequently, a smaller range of rip currents values is created. Outlying rip currents are lost as the max distance from the average decreases. The false positive rate is unaffected when projecting toward different components. This is a worst case scenario for a dataset as the negative images have less variance than the rip current samples. Projecting onto 1 component results in a major loss of descriptive information, which contributes to the inability to tell non-rip currents from rip currents.



(A) **(B)**

Figure 6. The false positive rates **(A)** and the detection rate **(B)** for each component separately projected as a threshold for classification.

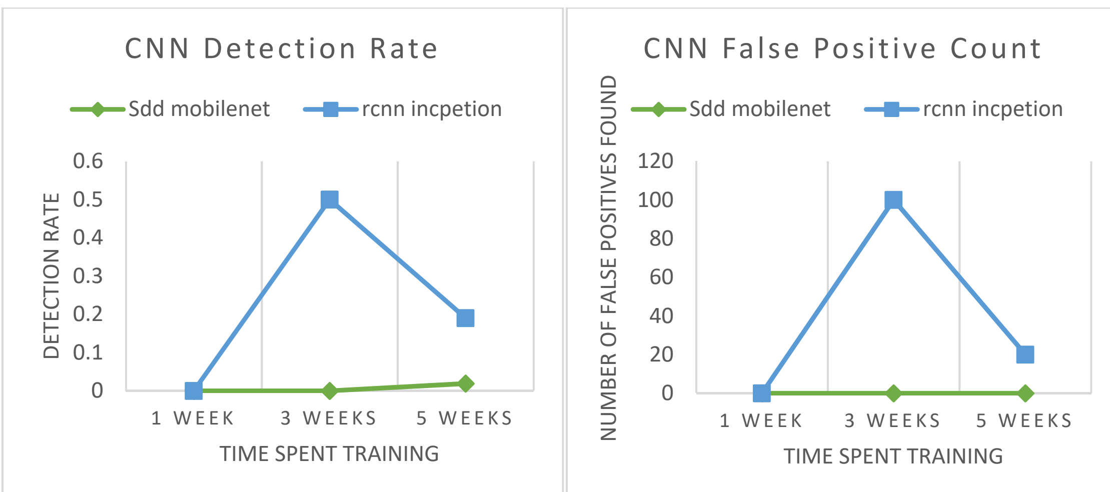
4.2 Support Vector Machines

The results for training the SVM with the average Haar feature vector is shown in Table 2. The SVMs accuracy reaches a max of 88% after scaling, optimization, and adding both circularity and black-white ratio. This is not a high enough accuracy rate as each layer of the Viola-Jones cascade can reach a detection rate of 99%. This supports the need for a better feature vector for training a rip current model. The 4% increase in accuracy support circularity and black-white ratio as viable features for rip currents. This is due to the semi-circular shape of rip currents and a similar orientation, which generates a similar black-white ratio for every sample.

Table 2. The results for adding different features and grid search for the SVM.

Change	**Accuracy of the SVM**
Average Haar features	74%
Circularity, black-white, and average Haar	78%
Grid search and scaling with all features	**88%**

4.3 Convolutional Neural Network



(A) (B)

Figure 7. Graph **(A)** shows the detection rates for the CNN during 1-5 weeks of training. Graph **(B)** shows the number of false positives found by the CNN during 1-5 weeks of training.

The results for the convolutional neural networks are shown in Figure 7. The detection rate of the Mobilnet model is absent until week 5 of training since it produces no predictions. The Mobilenet detection rate slightly increases after week 5. The false positive rate for the Mobilenet model remains at 0. The detection rate for the Inception model increases to 50% during week 3.The number of false positives found by the Inception model also increases to 100 during week 3. The detection rate and number of false positives for the Inception model drops to 18% and 20 false positives during week 5. The decrease in performance could be due to overfitting.

These results reflect the artificially created samples. There is not enough samples to properly train the CNNs as they require hundreds of thousands of samples. The warping of the positives seem to hurt performance for the Viola-Jones cascades. Therefore, creating many more artificial samples would not improve performance. Figure 8 shows a rip current put through the CNN detector. There are many imprecise detections around the rip currents in the surf zone.

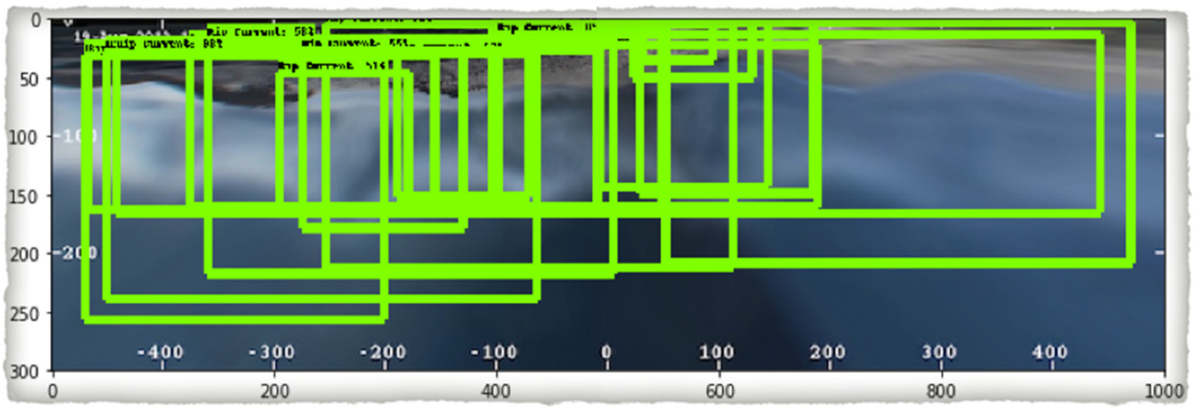


Figure 8. An image classified by the inception model after 3 weeks of training. The green boxes show detected rip currents in the image.

4.4 Viola-Jones

The results for each cascade are shown in Table 3. “Small Pos Non-Surf Neg”, “Created Non-Surf”, “Small Pos Surf Neg”, and “Created Surf” train on the created negative datasets. “Small Pos Small Neg” trains on the 24 by 24 rip current dataset and 24 by 24 negative surf zone samples [16]. “Created Surf” and “Created Non-Surf” train on created positive images [17]. The small dataset of rip currents and large dataset of any negative samples have the best tradeoff of detection rate and false positives found. A large amount of negative samples is helpful because smaller negative samples are down-sampled from them, which leads to a more robust detector.

Additionally, created positive samples seem to hurt the cascade performances. This could be due to the cascades learning the edges of warped samples instead of learning the rip current forms. Therefore, the models are overfitting to the dataset.

Table 3. The results for each Viola-Jones cascade.

Model	Det Rate	FP Count
Small Pos Non-Surf Neg	**0.88**	**15**
Created Non-Surf	0.63	6
Created Surf	0.63	6
Small Pos Surf Neg	1	75
Small Pos Small Neg	1	300

4.5 Meta-Classifier

This section first describes the results for adding new Haar features to the feature vector. Next, the accuracies for stacking are shown. Finally, the results for the meta-classifier are shown, which includes a comparison of each notable detector in the research.

4.5.1 Novel Features

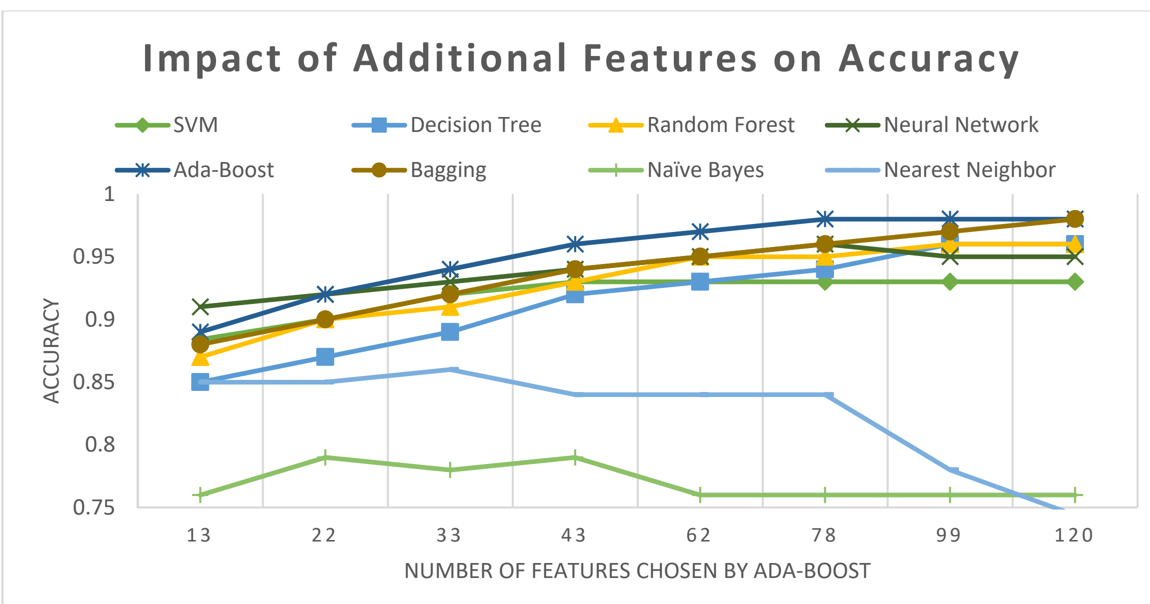


Figure 9. The results of continuously adding new features to the feature vector. The accuracy for each model is shown on the y axis while the number of features used is shown on the *x* axis.

Figure 9 displays the accuracies after continually adding new Haar features to the vector and re-evaluating each model. Ada-boost has a max performance after 77 Haar features are added to the feature vector. Ada-boost peaks first since the features are chosen by Ada-Boost. Consequently, they are biased towards it. The Haar features are optimized for splitting the dataset with a single value, which is what Ada-boost's weak classifiers do. Bagging, decision trees, and random forests are built on decision tree classifiers, which also split the data in a similar fashion. These features do not take advantage of the manner in which neural networks or SVMs learn. Therefore, they cannot match Ada-boost or bagging in performance. Nearest neighbors and Naïve Bayes suffer from adding additional features because they cannot handle larger dimensional features spaces.

4.5.2 Stacking

Table 4. The detection and false positive rates before and after stacking.

Model	Det. rate before stacking	Det. rate after stacking	FP rate before	FP rate after
Decision Tree	0.96	0.98	0.065	0.017
Naïve Bayes	0.89	0.97	0.37	0.034
Bagging	0.95	**0.99**	0.025	**0.011**
Ada-Boost	0.98	**0.99**	0.018	**0.012**
Nearest Neighbors	0.83	0.89	0.25	0.099
Random Forest	0.95	0.98	0.082	0.014
Neural Network	0.95	0.97	0.089	0.028
SVM	0.91	0.98	0.038	0.029

Table 4 shows the detection rate and false positive rate for each model's performance before and after stacking. Ada-Boost and Bagging both have the highest detection rates and lowest false positive rates after stacking because the features they train on give them an edge before stacking is applied. These models make decisions on the datasets by splitting on feature values. The Haar features chosen are optimized for splitting on this dataset, which gives them a higher initial performance than other classifiers.

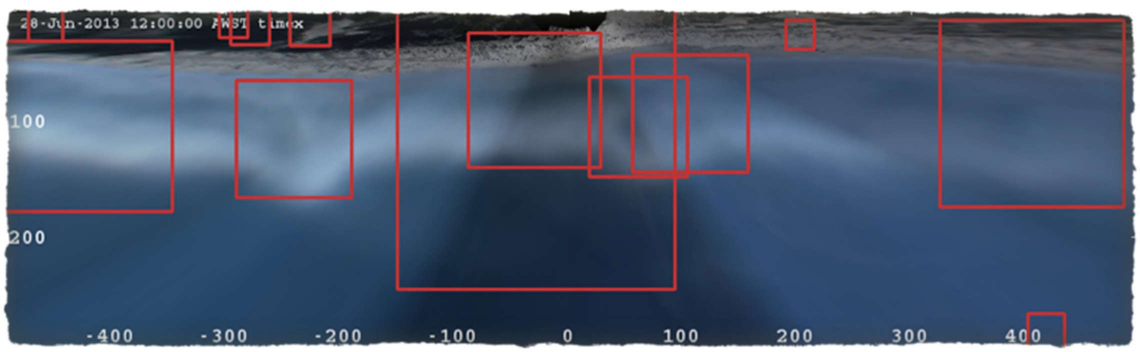
4.5.3 Meta-Classifier

Table 5. A comparison of Viola-Jones and the meta-classifier at different layers.

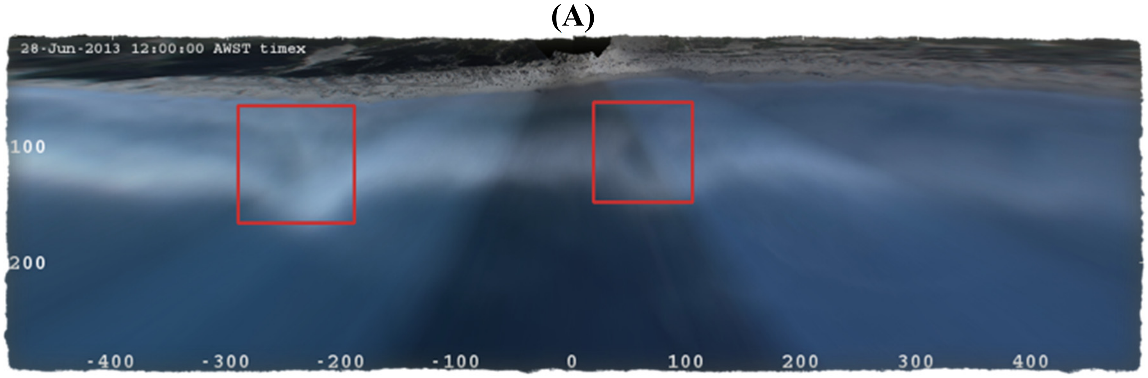
Layer	Viola-Jones Det.	Meta Det.	Viola-Jones FP count	Meta FP count
17	1	1	300	39
28	0.88	**0.85**	15	**8**
35	0.82	0.82	10	5
40	0.76	0.76	10	5

Table 6. A final comparison of methods.

Model	Det. Rate
CNN	0.5
Viola-Jones	0.88
Meta	**0.85**



(A)



(B)

Figure 10. Image **(A)** classified by Viola-Jones versus the bottom image **(B)** classified by the meta-classifier.

The meta-classifier compared to Viola-Jones at different layers is shown in Table 5. The meta-classifier improves the false positive performance at each layer it is applied. It is important to note that the meta-classifier at layer 28 provides a better performance than adding 12 layers to the cascade. Table 6 shows a final comparison of Viola-Jones, CNN, and the meta-classifier at optimum performance. The meta-classifier provides the best performance, if a lowest false positive rate is desired, by reducing the amount of false positives by 47%. CNNs have the lowest performance on the dataset in this comparison. This is, again, due to the nature of the data. Ada-Boost works better on smaller datasets than a CNN. Ada-Boost is also resistance to overfitting high variance datasets [15]. The large dataset training the CNNs is over fits the data because of the warped samples. The smaller dataset cannot train the CNNs adequately as they require a large amount of annotated samples. An image that has been run through the meta-detector and Viola-Jones detector is seen in Figure 10. The Viola-Jones image contains a larger number of false positives.

5. Conclusions

The rip current dataset seems to favor use with the Viola-Jones cascade and a meta-classifier back end. Ada-Boost and bagging have the best performance as the meta-classifier because of the Haar features. The new Haar features, in combination with the original 5 Haar features, are helpful for detection rip currents. Support vector machines are not accurate enough for classifying rip current images using averages of Haar features since they cannot match a layer of Viola-Jones. Convolutional neural networks need a larger, higher quality dataset to fully explore their potential.

This research makes a notable contribution to expert intelligent systems. It describes a detailed comparison of state-of-the-art detection models, which will provide researchers with some direction when choosing an algorithm for rip current identification. The study also presents meaningful features for extracting rip current data in future studies. The meta-classifier provides an improvement to the current state-of-the-art.

For future work, a large, annotated dataset should be developed for training convolutional neural networks. The resulting studies will generate a broader picture of CNN rip current detection capabilities.

Max distance from the average rip current image has the highest false positive rate of any classifier, but PCA could be useful as a method for feature reduction or rip current recognition in the future. The threshold could also be improved if more than one dimension generates the max distance from the average rip current image.

The features presented are optimized for Ada-boost, which provides a purpose for creating rip current features better suited for other machine learning algorithms in the future. The meta-learner detector is acceptable for automatic rip current detection, which will allow for more rip current climate studies to be conducted.

Supplementary Material

The code, used in this research work, is freely available here http://cs.uno.edu/~tamjid/Software/rip/code.zip and the datasets are available here [16] (small dataset) and here [17] (large dataset).

References

1. Castelle, B., et al., *Rip current types, circulation and hazard.* Earth-Science Reviews, 2016. **163**(Supplement C): p. 1-21.
2. Holman, R.A., et al., *Rip Spacing and persistence on an embayed beach.* Journal of Geophysical Research, 2006. **111**.
3. University, O.S. *Coastal Imaging Lab*. 2009 2009/01/05; Available from: cil-www.coas.oregonstate.edu.
4. Viola, P. and M.J. Jones, *Robust Real-Time Face Detection.* International Journel of Computer Vision, 2004. **57**(2): p. 137-154.
5. Howard, A.G., et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* 2017.

6. Szegedy, C., et al., *Going deeper with convolutions*, in *2015 IEEE Conference*. 2015: Boston, MA, USA.
7. Wolpert, D.H., *Stacked Generalization.* Neural Networks, 1992. **5**(2): p. 241-259.
8. *TensorFlow*. 2017; Develop]. Available from: https://www.tensorflow.org/tutorials/image_retraining.
9. Bradski, G., *The OpenCV Library.* Dr. Dobb's Journel of Software Tools, 2000.
10. Fabian Pedregosa, G.V., Alexandre Gramfort,Vincent Michel,Bertrand Thirion,Olivier Grisel,Mathieu Blondel,Peter Prettenhofer, and V.D. Ron Weiss, Jake Vanderplas,Alexandre Passos,David Cournapeau,Matthieu Brucher,Matthieu Perrot,Edouard Duchesnay, *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research. **12**: p. 2825-2830.
11. Gallop, S.L., et al., *Storm-driven changes in rip channel patterns on an embayed beach.* Geomorphology, 2011. **127**: p. 179-188.
12. Sebastian Pitman, S.L.G., Ivan D. Haigh, Sasan Mahmoodi, Gerd Masselink, Roshanka Ranasinghe, *Synthetic Imagery for the Automated Detection of Rip Currents.* Journal of Coastal Research, 2016. **75**: p. 912-916.
13. Bishop, C., *Pattern Recognition and Machine Learning*. Information Science and Statistics. 2009: Springer-Veriag New York.
14. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks.* 2012: p. 9.
15. Schapire, R.E., *Explaining AdaBoost*, in *Empirical inference, Festschrift in honor of Vladimir N. Vapnik*. 2013. p. 37-52.
16. Martan, C. and M.T. Hoque. *Dataset of rip current images* 2018 [cited 2018 May, 5th]; Available from: http://cs.uno.edu/~tamjid/Software/rip/dataset.zip.
17. Maryan, C. and M.T. Hoque. *Large dataset of rip current images*. 2018 [cited 2018 May, 5th]; Available from: http://cs.uno.edu/~tamjid/Software/rip/Large_dataset.zip.
18. Hastie, T., R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. 2009: Springer.
19. Maurya, A. *Support Vector Machines: How does going to higher dimension help data get linearly separable which was non linearly separable in actual dimension?* 2013; Available from: https://www.quora.com/Support-Vector-Machines-How-does-going-to-higher-dimension-help-data-get-linearly-separable-which-was-non-linearly-separable-in-actual-dimension.
20. Demirkaya, O., M.H. Asyali, and P.K. Sahoo, *Image Processing with MATLAB: Applications in Medicine and Biology*. 2008: CRC Press.
21. Pedregosa, F., et al. *sklearn.preprocessing.RobustScaler*. 2017 [cited 2018 May, 5th]; Available from: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html.