Applied Soft Computing Journal 78 (2019) 84–93

Contents lists available at ScienceDirect

Applied Soft Computing Journal

journal homepage: www.elsevier.com/locate/asoc

Machine learning applications in detecting rip channels from images

Corey Maryan [b,c], Md Tamjidul Hoque [a,b,*], Christopher Michael [c], Elias Ioup [c], Mahdi Abdelguerfi [a,b]

[a] Canizaro/Livingston Gulf States Center for Environmental Informatics, University of New Orleans, New Orleans, LA 70148, USA
[b] Department of Computer Science, University of New Orleans, New Orleans, LA 70148, USA
[c] US Naval Research Laboratory, Stennis Space Center, MS 39529, USA

HIGHLIGHTS

- Machine learning based effective and efficient rip channel detection from camera images.
- Design and development of novel Haar features for rip channel detection.
- Comparisons of machine learning algorithms for rip channel detection.
- Improving machine learning algorithms through stacking.

ARTICLE INFO

Article history:
Received 29 May 2018
Received in revised form 4 January 2019
Accepted 10 February 2019
Available online 13 February 2019

Keywords:
Rip channel
Deep learning
Support vector machine
Principal component analysis
Meta-learner

ABSTRACT

Images containing rip channels are used in oceanographic studies and can be preprocessed for these studies by identifying which regions of the image contain rip channels. For thousands of images, this process can become cumbersome. In recent years, object detection has become a successful approach for identifying regions of an image. There are several different algorithms for detecting objects from images, however, there is no guidance as to which algorithm works well for detecting rip channels. This paper aims to compare and explore state-of-the-art machine learning algorithms, including the Viola–Jones algorithm, convolution neural networks, and a meta-learner on a dataset of rip channel images. Along with the comparison, another objective is to find suitable features for rip channels and to implement the meta-classifier for competition with the state of the art. The comparison suggests the meta-classifier is the most promising detection model. In addition, five new Haar features are found to successfully supplement the original Haar feature set. The final comparison of these models will help guide researchers when choosing an appropriate model for rip channel detection, the new Haar features provide researchers with valuable data for detecting rip channels, and the meta-classifier provides a method for increasing the accuracy of a detector through classifier stacking.

1. Introduction

Rip currents are narrow currents that flow seaward through the surf zone of a beach [1] and are a significant hazard to swimmer safety [2]. Rip currents are caused by alongshore variations in wave breaking. In many cases, rip currents are strong enough to transport seafloor sediments, creating a bathymetric low, or rip channel, through the surf-zone sandbar. Some rip channels are readily visible in processed video imagery, and serve as a proxy for the location of potentially dangerous rip currents, making rip channels a popular topic of investigation [3]. Rip channels are generally identified with time-exposure images as instantaneous snapshots do not provide enough contrast between rip channels and sandbar crests [4]. These time exposure images reveal a specific type of rip channel known as a bathymetrically-controlled channel [1]. Since this type of rip channel is exposed through images, they will be the focus of the study. Time exposure images are created by time averaging video frames over a user-defined period, frequently 10 min for coastal imagery. A full description of the technique for creating time exposure imagery in the coastal area is available in [3]. Time exposure images reveal the location of rip channels as a region of dark pixels, where waves are not breaking, between bright pixels associated with foam generated by wave breaking over a surf-zone sandbar (Fig. 1). Time exposure images can be orthorectified by the process described in [5,6] to create a map view of the beach. Then, the location of rip channels can be identified. While the location of rip channels are an important aspect of oceanographic research [7], the process of identifying rip

[*] Corresponding author at: Canizaro/Livingston Gulf States Center for Environmental Informatics, University of New Orleans, New Orleans, LA 70148, USA.
E-mail addresses: cmaryan@uno.edu (C. Maryan), thoque@uno.edu (M.T. Hoque), chris.michael@nrlssc.navy.mil (C. Michael), Elias.Ioup@nrlssc.navy.mil (E. Ioup), mahdi@cs.uno.edu (M. Abdelguerfi).

https://doi.org/10.1016/j.asoc.2019.02.017
1568-4946/© 2019 Elsevier B.V. All rights reserved.

channels in time exposure imagery is labor intensive, and there has been little research done in automatically detecting the location of rip channels from images. With better observations of rip channels, the fundamental understanding of rip channel behavior will be improved. One possible solution to this problem is to apply object detection algorithms to each image.

Object detection has become a popular approach to find certain regions within an image [9–11]. Object detection methods are applied to a wide variety of objects, which makes applying such techniques to rip channel images intuitive. There are several popular detection algorithms to choose from in the literature, such as convolutional neural networks or Viola–Jones, yet, there is little research supporting which algorithm is appropriate for channel rip-current detection. A comparison of state-of-the-art models should determine an effective and efficient 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 channel image using principal component analysis (PCA), and the meta-classifier [12]. The Viola–Jones algorithm was chosen because of its efficiency and resistance to overfitting. Deep learning is a popular solution to many object detection problems because of its utilization of modern computing power. Convolution neural networks, in particular, have the ability to create their own feature data instead of the user supplying data. SVM's tend to have high performance with high dimensional data, which images naturally contain. PCA is a technique that is typically utilized for image recognition, not detection. However, it can identify which regions of an image are more unique to that specific image than other regions, which is useful for finding detectable image features.

This paper aims to compare state-of-the-art object detection models on rip channels, which will guide researchers in choosing an algorithm for detecting rip channel location from images. The Haar features applied by the Viola–Jones algorithm are optimized for faces. Therefore, this paper also aims to create Haar features like the Viola–Jones features but optimized for rip channel detection. The last objective is to provide a competitive model for rip channel detection through a technique called classifier stacking.

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 specific implementation details of each model. Section 5 shows the results of the experiments. Section 6 contains an overall discussion of the pros and cons based on the methods. Finally, Section 7 discusses the conclusions.

2. Background

This section contains a background of rip channels and object detection research fields.

2.1. Rip channels

Time exposed images constrain sandbar and rip channel locations [4]. The rip channels identified in this type of imagery are usually channel rip-currents or bathymetrically controlled rip currents. Rip channels are usually studied in situ [13] or with imagery [14–16]. The in-situ methods conduct a sonar survey of bathymetry. These data can be processed to determine the locations or morphology of rip channels. However, the cost and effort associated with surveying rip channels are often prohibitive. Naturally, studies involving imagery substitute these tools for the imagery to gather data. The primary focuses of study for rip channels when using imagery are morphology, or shape, [15] and location [1,14,16]. Some studies involving rip channel morphology have applied computer-based techniques to locate rip channels using semi-automated algorithms with expert corrections to find maxima and minima pixel intensity values [15]. Rip channels tend to appear darker than the rest of the surf zone, which is why pixel intensity is a popular approach. Other studies involve finding rip channel location [14,16]. Each of the mentioned studies involves identifying rip channels through imagery, however, only [15] attempts an automated approach, and it is semi-automated. There is a clear lack of research applying machine learning to create fully-automated methods for finding rip channel locations in imagery. Finding a fully-automated method has possible applications for object detection.

2.2. Machine learning based object detection

Machine learning employs algorithms that fit data to a model [17]. 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 a 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 [10,11,18]. Object detection has numerous algorithms to choose from because of the wide variety of machine learning techniques in the literature. Therefore, a comparison of such algorithms is valuable when deciding which one is most appropriate for rip channel detection. Rip channels are a lack of focus for object detection, which also creates a critical need for adequate rip channel 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 the detection rate and false positive rate. These metrics represent how accurately a model can identify rip channels and avoid misclassifying non-object samples, respectively. Examples of popular object detection algorithms are Viola–Jones [9] and convolutional neural networks [10,11,18].

Viola–Jones has success in the area object detection, specifically with faces [9]. This is due to its speed and robust nature. Viola–Jones employs the Ada-boost algorithm, which is highly resistant to overfitting [19]. 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 rate 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 near instantaneous evaluation speed.

Convolutional neural networks are also successful in detecting many types of objects [10,11,18]. These networks create their own features for detecting objects, which are based on image filters. Over thousands of samples, 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 hundreds of thousands of samples to train on because they overfit easily on a small dataset due to being formulated as high order polynomials for flexibility.

3. Methodology

This section describes how each detector is set up and run for comparison, including max distance from the average rip channel image, SVM, convolutional neural networks, Viola–Jones, and the meta-classifier. The detectors are compared on a test set of rip channel images with detection rate and false positive count. All metrics are generated using 10-fold cross-validation.

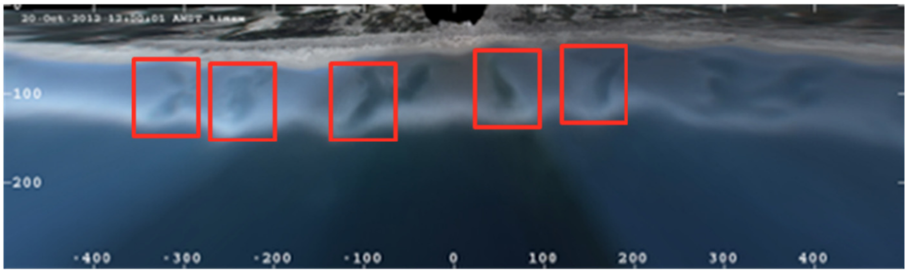


Fig. 1. An example image of the Secret Harbour, Australia shoreline [8]. The rip channels are seen in red boxes. The numbers around the borders of the image indicate meters from the camera.

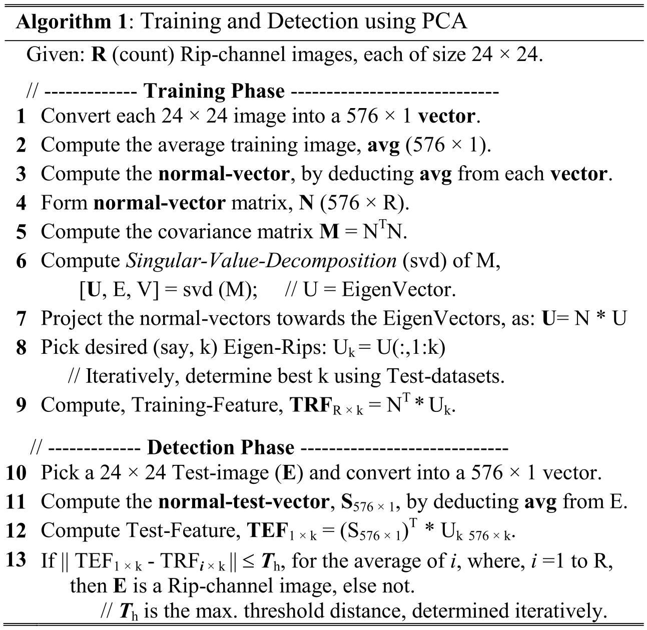
3.1. Dataset

The dataset contains 514 rip channel examples, including the test dataset [20]. These samples are all channel rip currents, that is, bathymetrically-controlled rip channels. These rip channels are 24 pixel by 24 pixel images extracted from 1334 pixel by 334 pixel images. Larger images are downloaded from the backlog of beach imagery on the Oregon State University website [8]. The images are timex images that have been orthorectified with the process mentioned in Section 2.1, which gives them a map view appearance. Images are a time average of 1200 frames collected at 2 Hz over 10 min. Images average about 2 to 4 rip channels per image. In total, are 102 images of shorelines that contain the extracted rip channel samples. Small rip channel samples are extracted with the GIMP image editor. The entire rip channel is extracted from the image, not just one aspect of the channels in particular. This is done because object detection algorithms typically take the entire object and the small region around the object as input. 3 to 4 timex image are taken per day, during daylight hours, by the cameras located at Duck, North Carolina and Secret Harbour, Australia. Rip channels occur frequently at these two sites. An example rip channel image from these sites is seen in Fig. 1. Fig. 1 contains 5 rip channels indicated by red rectangles. These types of images are given, as input, to the detectors. The rip channel samples, extracted from these large images, are all normalized to 24 pixels by 24 pixels for standard Haar feature calculation. This is done since 24 pixels by 24 pixels is the smallest possible input size of an image. In the GIMP editor, images that are not 1:1 are scaled using cubic interpolation. After being reduced to 24 by 24, the samples are also converted to grayscale. This is done to help minimize the effect of different lighting effects on the detection models. 10-fold cross validation splits the dataset accordingly for evaluation, where each fold is 10% of the dataset. The models train on the remaining 90% and evaluate on some fold. This is repeated until each fold has been evaluated on.

Convolutional neural networks used in this study need to train on many thousands of images to attain decent results [18]. Therefore, creating a larger dataset is necessary. The process of data augmentation can create new, positive samples by applying distortions to every sample. These samples can then impose onto a larger background image. A warp factor of 0.1 creates a dataset of 4000 rip channel images [21]. This process can also create more negative samples in a similar fashion. This dataset is compared with the small dataset since the models can accept either as input. A comparison of datasets helps determine the most effective type of input for each model.

3.2. Threshold from the average rip channel image using PCA

The process of object detection is simplified if a range of values representing rip channels is found. Normally, a model must train on a dataset of positive and negative samples. Eliminating the negative samples simplifies the size of the dataset. To this end, principal component analysis (PCA) is employed [22]. An example of the method (Algorithm 1) is shown in Fig. 2. This method reduces the number of dimensions in the 24 by 24 image dataset of a rip channel into a number of chosen components from the Eigenvector. Each component is projected upon to generate a max distance (threshold, $\boldsymbol{T}_h$) from the average rip channel 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 channel vector upon 1 component generates a rip channel feature vector of size 1. Finding the greatest distance between any feature vector and the average feature vector creates the max distance ($\boldsymbol{T}_h$) from the average. This is applied as a threshold for rip channel identification. If a new data sample, projected upon a component, generates a feature vector with a distance less than the max distance ($\boldsymbol{T}_h$), then it is classified as a rip channel. Otherwise, it is classified as not a rip channel. The thresholds are tested for their detection rate and false positive count with 10-fold cross-validation.



Algorithm 1: Training and Detection using PCA

```
 Given: R (count) Rip-channel images, each of size 24 × 24.
 // ------------- Training Phase -----------------------------
 1  Convert each 24 × 24 image into a 576 × 1 vector.
 2  Compute the average training image, avg (576 × 1).
 3  Compute the normal-vector, by deducting avg from each vector.
 4  Form normal-vector matrix, N (576 × R).
 5  Compute the covariance matrix M = $N^T N$.
 6  Compute Singular-Value-Decomposition (svd) of M,
        [U, E, V] = svd (M);     // U = EigenVector.
 7  Project the normal-vectors towards the EigenVectors, as: U= N * U
 8  Pick desired (say, k) Eigen-Rips: $U_k$ = U(:,1:k)
        // Iteratively, determine best k using Test-datasets.
 9  Compute, Training-Feature, $TRF_{R \times k}$ = $N^T * U_k$.

 // ------------- Detection Phase -----------------------------
 10 Pick a 24 × 24 Test-image (E) and convert into a 576 × 1 vector.
 11 Compute the normal-test-vector, $S_{576 \times 1}$, by deducting avg from E.
 12 Compute Test-Feature, $TEF_{1 \times k}$ = $(S_{576 \times 1})^T * U_{k\ 576 \times k}$.
 13 If || $TEF_{1 \times k}$ - $TRF_{i \times k}$ || ≤ $T_h$, for the average of i, where, i =1 to R,
        then E is a Rip-channel image, else not.
            // $T_h$ is the max. threshold distance, determined iteratively.
```

Fig. 2. A depiction of the max distance (T_h) from the cluster of positive images classify a given image.

3.3. Support vector machines

Support vector machines require the user to provide features for training. The Haar feature space contains over 200,000 possible features for a 24 pixel by 24 pixel window. This is a descriptive set with success in object detection, but over 200,000 dimensions

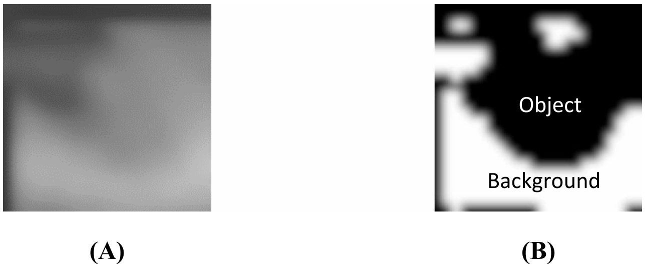


Fig. 3. Image **(A)** is an example of a rip channel sample in grayscale. Image **(B)** is a segmented version of the same image.

Table 1
Hyperparameters for the CNNs.

Model	Inception	Mobilenet
Activation function	Softmax	Rectifier
Initial weights	Std. dev 0.01	Std. dev 0.03
Batch size	1	24
Dropout for regularization	–	0.8
Learning rate	0.0003 at 0 iterations 0.0003 at 900000 iterations 0.00003 at 1200000 iterations	0.004 at 0 iterations Decay factor 0.95
Momentum	0.9	0.9

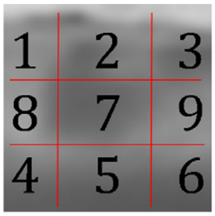


Fig. 4. The matrix of the new features laid on top of a rip channel image. Each number represents an area of space that can be extracted from the integral image to use in a Haar feature formula.

are too many 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 [23] and black–white ratio are added to the feature vector for each sample. Circularity is taken from the segmented image by finding all connected components in the binary image. An example is shown in Fig. 3. The rip channel object is in black while the background is in white. The equation for circularity is shown in (1).

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

In Eq. (1) , 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-diagonals but topologically adjacent pixels. This circularity-measurement (1), assumes that the object is in black and the background is in white.

The black–white ratio is taken from an image of a rip channel where each pixel is converted to totally black or totally white, depending on a threshold. The black–white ratio is chosen as a feature based on the intuition that similar, normalized objects will have a similar black–white ratio.

3.4. Convolutional neural networks

Recently, convolutional neural networks have become a popular method for deep learning and object detection [10,11,18]. The CNNs chosen are the "Mobilenet" and "Inception" models. The Mobilenet model is geared more toward speed [10] while Inception is geared more toward accuracy [11]. These networks are chosen as the first steps in evaluating CNN capabilities on detecting rip channels. The models use hyperparameters defined in Table 1.

The Inception model trains for 156 284 iterations while the Mobilenet model train for 442 691 iterations. They train on the augmented dataset as they require thousands of images. Note that, they each train for five weeks, but one model simply iterates faster. The models then run 10-fold cross-validation on the set of rip channel images. A common approach to CNN is to use pre-trained models to build a network and simply fine-tune it to the desired object. An explanation for why this approach is not taken is discussed in Section 4. Overall, the research involving the CNN's is still in a very preliminary stage and, though the results of the initial testing are presented, will not be compared directly to the other methods.

3.5. Viola–Jones method

A combination of different images creates a set of Viola–Jones cascades. There are 5 total cascades created. The first cascade trains on the small dataset of rip channels and negative images [20]. The second cascade is built with the small rip channel dataset and a large dataset of surf zone negatives. The third cascade is built from small rip channel images and a large negative dataset of any image. The fourth cascade is built with created negatives of the surf zone and the larger positive image dataset. The last cascade is built with created negatives of any image and the large created dataset of rip channels [21]. The cascades are run on the test set of images after they train to completion. A false positive threshold of 0.7 per layer and detection threshold 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 developed.

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 channels. 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 Fig. 4. In Fig. 4, each number of the matrix corresponds to a region of intensity values in the image. The average rip channel 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 Fig. 4), respectively. Regions 2 and 7 help create the most accurate Haar feature for rip channels. 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 2. The formula for each feature in Table 2 is found by adding some combination of integral image regions from Fig. 4 together then subtracting other regions from the total.

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. AdaBoost finds the most appropriate Haar features for rip channels from both the original (see Fig. 5) and the new features (see Fig. 6).

Table 2
The results of using each feature to train 10-layer cascades of weak classifiers. The Formula column refers to the matrix described in Fig. 4.

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.

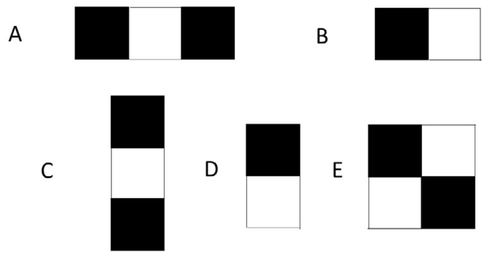


Fig. 5. Example Haar features (A) The "Three Horizontal" feature, (B) The "Two Horizontal" feature, (C) The "Three Vertical" feature, (D) The "Two Vertical" feature, (E) The "Four" feature [9].

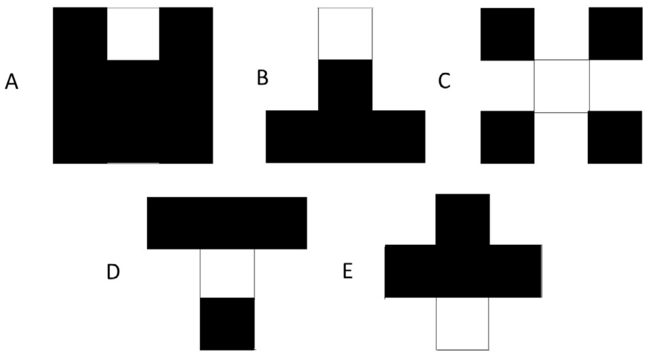


Fig. 6. Rip Channel Features: (A) "Three columns", (B) "Inverted T_1", (C) "X", (D) "T_1" and (E) "Cross".

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 detection rate levels off for each basic model. The distribution for the final feature vector that trains the meta-classifier is 70% face features 30% rip channel features.

3.6.2. Meta-Classifier

The meta-classifier trains on the confidence of previous models. Confidence values from other models alleviate what a model may lack [12] in terms of performance. The basic models include SVM, neural network, decision tree, random forest, k-nearest neighbors, Naïve Bayes, bagging, and Ada-boost. The first 77 Haar features, chosen by Ada-boost, train each basic classifier. 10-fold cross validation yields the probability of the models. Every rip channel sample has a training vector of 85 after adding the 8 output confidence values from the basic models. The 85-feature vector trains the meta-classifier. Each model that generates confidence values for training the meta-classifier is also 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. Therefore, this further reduces the number of false positives while attempting to keep the same detection rate.

4. Implementation

The max distance from the average rip channel image is generated and applied for classification through (PCA) in Matlab. The Matlab routine*svd* (singular value decomposition) is run to generate the Eigenvector.

The Scikit-learn package in Python runs the SVM [24]. This contains its own grid search method to find the optimal parameters. Matlab is used to generate the circularity and black–white ratio features. The Matlab routine *bwconncomp* generates the segmented images from the binary images, which are used to calculate circularity. 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. These images help calculate the black–white ratio feature. The SVM uses a radial basis function (RBF) kernel. Grid search, inside the Scikit-learn package, optimizes the parameters, *C* and *gamma*, for the kernel. The results of the grid search for the small dataset of rip channels are $C = 4.0$ and $gamma = 0.00390625$. A robust scaler object [25] from the Scikit-learn package is applied to the rip channel 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 convolutional neural networks in this study are built from the TensorFlow framework [26], which is developed by Google. TensorFlow's default configurations build both the Inception and Mobilenet models. Pre-built configurations and the framework make it easier to generate results since parameters are pre-defined. It is possible to retrain a network instead of building one from scratch. However, this required preprocessing the images in a more complex manner. Due to time constraints, this route was not chosen but would be useful to explore later. The parameters are not fine-tuned for the dataset for the same reason. As stated in

Table 3
A comparison of methods when searching for an image for rip channels.

Method	Detection rate	False positive count
Max distance from average	1.0	>400
SVM	0.98	>400
Viola–Jones	0.88	15
Meta-Learner	**0.85**	**8**

Table 4
The results for adding different features and grid search for the SVM.

Change	Detection rate of the SVM
Average Haar features	81%
Circularity, black–white, and average Haar	85%
Grid search and scaling with all features	**95%**

Table 5
The results for each Viola–Jones cascade.

Model	Detection 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

Table 6
Training the meta-classifier on different types of Haar features.

Types of Haar features training meta-classifier	Detection rate	False positive count
Original face features only	0.83	12
Rip channel features only	0.84	14
Both	**0.85**	**8**

Section 3, the research done using these networks is preliminary and would not be a fair comparison with the other methods in its current state. Therefore, the results of applying the networks with pre-defined parameters are shown but are not directly compared to the other methods.

The OpenCV package [27] has an implementation of the Viola–Jones algorithm for use with any object. This package in Python builds the cascades using *opencv_traincascade*. The OpenCV package is also used for creating new samples described in Section 3.1.

Most of the Scikit-learn [24] models are used as basic classifiers to generate confidence values for each rip channel sample. Each model except for SVM run with the default Scikit-learn parameters. The SVM parameters are described in paragraph 2 of this section.

5. Results

This section describes the false positive count and detection rate for each of the previously mentioned methods of comparison except CNN's. Table 3 shows a comparison of Viola–Jones, SVM, max distance from the average, and the meta-classifier at optimum performance. The meta-classifier provides the best performance if the lowest false positive rate is desired, by reducing the number of false positives by 47% from Viola–Jones. Max distance from the average has the lowest performance on the dataset in this comparison.

The results for max distance from the average using PCA are shown in Fig. 7. Here, the detection rate per component of the Eigenvector decreases toward the back of the vector. The false positive counts remain unaffected.

The results for training the SVM with the average Haar feature vector is shown in Table 4. The SVMs detection rate reaches a max of 95% after scaling, optimization, and adding both circularity and black–white ratio.

The results for the convolutional neural networks are shown in Figs. 8 and 9. The detection rate of the Mobilenet model is absent until 221 345 iterations of training since it produces no predictions. The Mobilenet detection rate slightly increases to 0.019 after this point. The false positive count remains at 0. The detection rate and the number of false positives for the Inception model increase to 0.5 and 100, respectively, after 118713 iterations. Then, the detection rate and false positive count steadily decrease as the simulation reaches 158 284 iterations. An example image run through the Inception model can be seen in Fig. 10.

The results for each cascade are shown in Table 5. “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 channel dataset and 24 by 24 negative surf zone samples [20]. “Created Surf” and “Created Non-Surf” train on created positive images [21]. The small dataset of rip channels and the large dataset of any negative samples have the best tradeoff of detection rate and false positives found.

Table 6 displays the results of training the meta-learner on different Haar features. Applying both face and rip channel Haar features gives the highest benefit to the meta-learner. Table 7 shows the detection rate and false positive count for each model before and after stacking. Ada-Boost and Bagging both have the highest detection rates and lowest false positive counts after stacking.

The meta-classifier compared to Viola–Jones at different layers is shown in Table 8. 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 better performance than adding 12 layers to the cascade. An image that has been run through the meta-detector and Viola–Jones detector is seen in Fig. 11. The image runs through Viola–Jones contains a larger number of false positives.

6. Discussion

This section discusses the advantages and disadvantages of each method based on the results found.

6.1. Threshold from the average rip channel image using PCA

Threshold from the average rip channel has the benefit of not needing to train a model. An average is taken and compared again any potential sample. In addition, PCA can generate the features needed from the image, which avoids having to design features.

Projecting onto 1 component results in a major loss of descriptive information, which contributes to the inability to tell non-rip channels from rip channels.

6.2. SVM

SVMs tend to work well with high dimensional data [28]. They also have a very high detection rate on the dataset. The shortcomings start with parameter requirements.

In order to optimize the SVM, its optimal parameters must be found with grid search, similar to CNNs. This can take a great deal of time to complete. In addition, SVMs lack any kind of cascading architecture. This gives them a disadvantage when classifying windows that do not contain rip channels. This is a rather small dataset of rip channels, which implies a small dataset of negative images to train on. It is impossible to differentiate rip channels from all possible negative images that may appear with one trained SVM since can be billions of negative sample possibilities to classify.

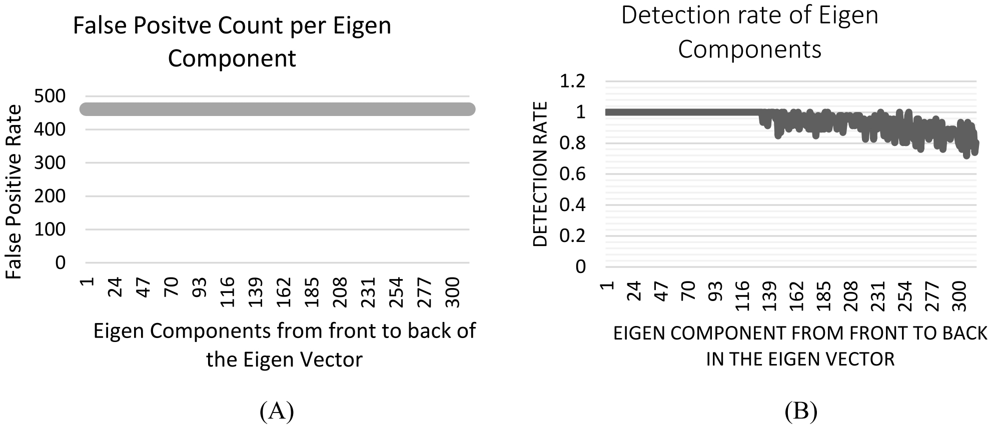


Fig. 7. The false positive counts **(A)** and the detection rate **(B)** for each component separately projected as a threshold for classification.

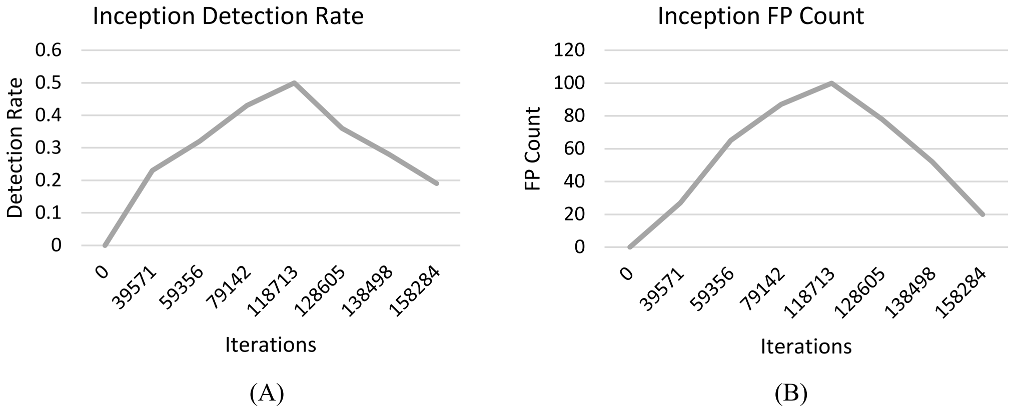


Fig. 8. Graph **(A)** shows the detection rates for the Inception model during training. Graph **(B)** shows the number of false positives found by the Inception model during training.

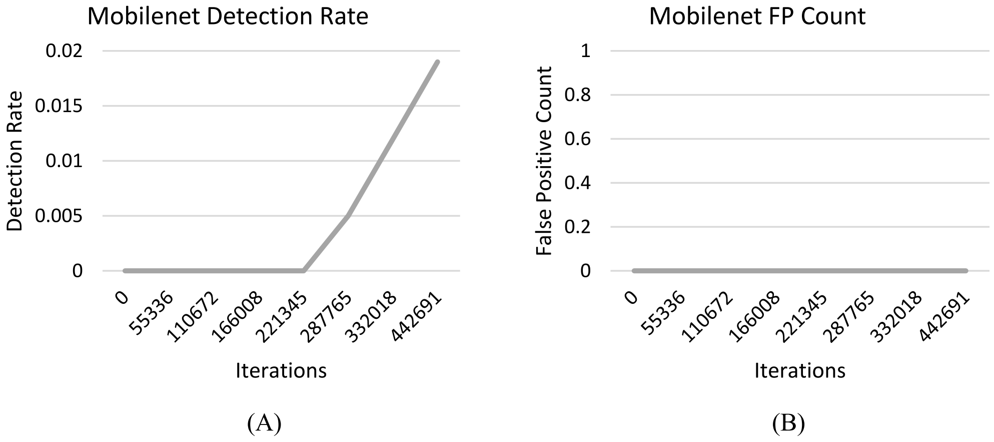


Fig. 9. Graph **(A)** shows the detection rates for the Mobilenet model during training. Graph **(B)** shows the number of false positives found by the Mobilenet model during training.

6.3. CNN

CNNs have the advantage of generating their own features through the use of feature maps.

They learn which filters are meaningful to the object in question through training and convolution. CNNs are highly customizable through parameters and a capable of detecting multiple objects in the same image.

Their disadvantage is how easily they overfit to a dataset. Though the results could reflect training on artificially created samples, the hyperparameters are not fine-tuned for rip channels. CNN's take a long time to train from scratch and longer to validate the hyperparameters needed to optimize them. There are not enough samples to properly train the CNNs as these models require hundreds of thousands of samples. Although the Inception model is able to identify some regions with rip channels, it is imprecise in its detections.

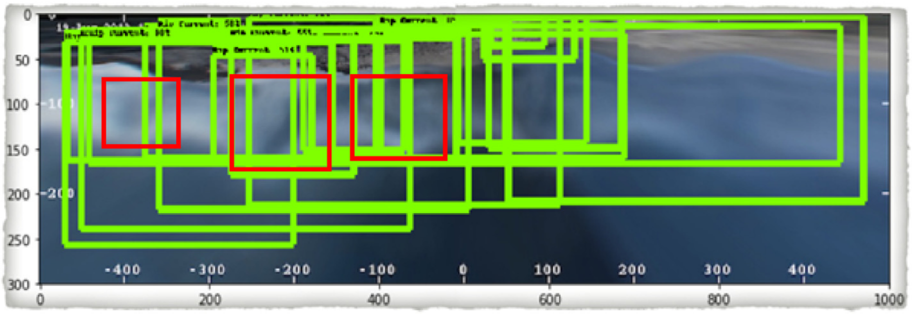
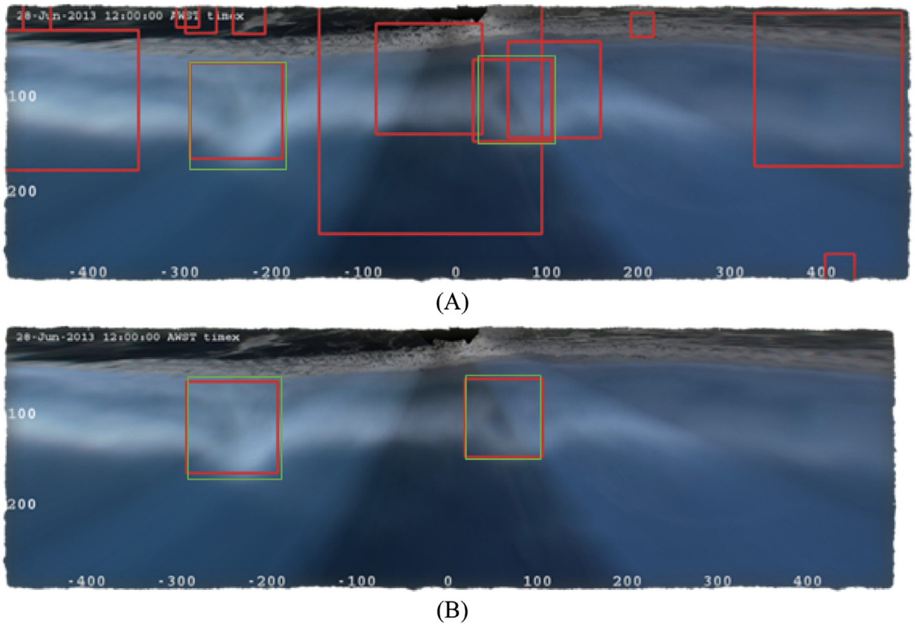


Fig. 10. An image classified by the inception model after 3 weeks of training. The green boxes show possible rip channels predicted by the CNN in the image. The red boxes show the actual rip channels.



(A)

(B)

Fig. 11. Image **(A)** shows standalone Viola–Jones and in (**B**) the same Viola–Jones model with the meta-learner classifying at the back-end. The green boxes indicate the ground truth.

Table 7
The detection rate and false positive counts before and after stacking. In the after case, the named method is used as the Meta-learner and the rest as the base-learner, whereas the before case is the standalone version.

Model	Det. rate before stacking	Det. rate after stacking	FP count before	FP count after
Decision tree	0.96	0.98	32	8
Naïve Bayes	0.89	0.97	185	17
Bagging	0.95	**0.99**	12	**5**
Ada-Boost	0.98	**0.99**	9	**6**
Nearest neighbors	0.83	0.89	125	49
Random forest	0.95	0.98	41	7
Neural network	0.95	0.97	44	14
SVM	0.91	0.98	19	14

6.4. *Viola-Jones*

Viola–Jones employs cascading architecture. This allows the maximum amount of processing power to spend only on windows that are potentially rip channels. All negative windows are thrown out in earlier layers. Viola–Jones is fairly quick to train and takes less time than a deeper learner, such as a CNN. There are very little parameters to specify, which is both beneficial and detrimental. Beneficial in that there is no time needed to grid search for optimal parameters but detrimental in that Viola–Jones is less customizable. Ada-Boost, the learning algorithm of Viola–Jones, is also resistance to overfitting high variance datasets [19]. This is the main reason Viola–Jones is the best out of the box classifier on the rip channel dataset.

Viola–Jones is highly dependent on the Haar features. If a data sample cannot be classified with them, then it makes Viola–Jones ineffective. Also, the detection rate will continually drop as more layers are added to the cascade which puts a limit on the number of possible layers. Finally, Viola–Jones is not able to detect more than one object in an image at once.

Table 8
A comparison of Viola–Jones and the meta-classifier to reclassify the output of Viola–Jones, to show that combination can help reduce the required layers in Viola–Jones.

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

6.5. *Novel Haar features*

The Haar features presented have a higher detection rate than the Viola–Jones features. This is plausible since they are built from the average rip channel just as the Viola–Jones Haar features are built from the average face. It is interesting, however, that combining the 2 sets results in the highest performance. This is most likely due to the nature of the features. Each of these features is of a different type. Each type contains many thousands of features. This means giving Ada-boost a larger feature space allows it to find some of each type that works better on the rip channel dataset than the rest, regardless of which set they are from.

6.6. *Meta-Learner*

The meta-learner is added to the back of the Viola–Jones cascade, giving it the same advantages and disadvantages. Additionally, the meta-learner has the benefit of choosing a different final classifier, which gives a bit more customization than Viola–Jones. The meta-learner can achieve the same false positive count as a cascade of Viola–Jones with a high number of layers, but without needing the extra layers. This increases the overall detection rate for the cascade but reduces the number of false positives.

The meta-learner is not as intuitive to use as the other models and requires much more input than the other models. Since it contains a high number of machine learning models, it also needs many parameters defined for each model.

7. Conclusions

The rip channel 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 way Haar features divide the dataset, just as a decision tree would. The new Haar features, in combination with the original 5 Haar features, are helpful for detection rip channels as they have improved performance of the meta-learner more than the original 5 alone. Support vector machines are not accurate enough to classify rip channel images using averages of Haar features since they do not have cascading architecture like Viola–Jones. Additional research is needed to explore the potential of convolutional neural by fine-tuning hyperparameters and through the use of pre-training and checkpoints.

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 channel identification. The study also presents meaningful features for extracting rip channel data in future studies. The meta-classifier provides an improvement to the state-of-the-art.

These results indicate three possible approaches for future work. A large dataset could be developed for training convolutional neural networks. Due to time constraints, proper validation of hyperparameters could not be done nor making use of checkpoints and network pretraining. Pretrained CNNs should be run on the dataset and hyperparameters validated. The resulting studies will generate a broader picture of CNN rip channel detection capabilities. A model fine-tuned for this specific dataset will probably have better results.

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

The features presented are optimized for Ada-boost, which provides a purpose for creating rip channel features better suited for other machine learning algorithms in the future. The meta-learner detector is acceptable for automatic rip channel detection, which will allow for more rip channel 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 [20] (small dataset) and here [21] (large dataset).

Acknowledgments

MTH gratefully acknowledges the Louisiana Board of Regents through the Board of Regents Support Fund LEQSF (2016–19)-RD-B-07.

References

[1] B. Castelle, et al., Rip current types, circulation and hazard, Earth-Sci. Rev. 163 (Suppl. C) (2016) 1–21.
[2] V.A. Gensini, W.S. Ashley, An examination of rip current fatalities in the United States, Nat. Hazards (2009).
[3] R.A. Holman, J. Stanley, The history and technical capabilities of argus, Coast. Eng. 54 (6) (2007) 477–491.
[4] T.C. Lippman, R.A. Holman, Quantification of sandbar morphology, J. Geophys. Res. 94 (1989) 995–1011.
[5] K.T. Holland, et al., Practical use of video imagery in nearshore oceanographic field studies, IEEE J. Ocean. Eng. 22 (1) (1997).
[6] J. Heikkilä, O. Silvén, A Four-Step Camera Calibration Procedure with Implicit Image Correction, 1997.
[7] R.A. Holman, et al., Rip spacing and persistence on an embayed beach, J. Geophys. Res. 111 (2006).
[8] O.S. University, Coastal imaging lab, 2009, 2009/01/05; Available from:http://cil-www.coas.oregonstate.edu.
[9] P. Viola, M.J. Jones, Robust real-time face detection, Int. J. Comput. Vis. 57 (2) (2004) 137–154.
[10] A.G. Howard, et al., MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, 2017.
[11] C. Szegedy, et al., Going Deeper with Convolutions, 2015.
[12] D.H. Wolpert, Stacked Generalization.
[13] J. MacMahan, et al., Mean Lagrangian flow behavior on an open coast rip-channeled beach: A new perspective, Mar. Geol. 268 (2010) 1–15.
[14] R. Ranasinghe, G. Symonds, R. Holman, Quantitative characterisation of RIp dynamics via video imaging, Coast. Sediments 2 (1999).
[15] S.L. Gallop, et al., Storm-driven changes in rip channel patterns on an embayed beach, Geomorphology 127 (2011) 179–188.
[16] S.L.G. Sebastian Pitman, Ivan D. Haigh, Sasan Mahmoodi, Gerd Masselink, Roshanka Ranasinghe, Synthetic imagery for the automated detection of rip currents, J. Coast. Res. 75 (2016) 912–916.
[17] C. Bishop, Information Science and Statistics, Springer-Veriag, New York, 2009.
[18] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, 2012, p. 9.
[19] R.E. Schapire, Explaining adaboost, in: Empirical Inference, Festschrift in Honor of Vladimir N. Vapnik, 2013, pp. 37–52.
[20] C. Martan, M.T. Hoque, Dataset of rip current images, 2018, [cited 2018 May, 5th]; Available from: http://cs.uno.edu/~tamjid/Software/rip/dataset.zip.
[21] C. Maryan, 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.
[22] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, Springer, 2009.

[23] O. Demirkaya, M.H. Asyali, P.K. Sahoo, Image Processing with MATLAB: Applications in Medicine and Biology, CRC Press, 2008.

[24] G.V. Fabian Pedregosa, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, V.D. Ron Weiss, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[25] F. Pedregosa, et al., Sklearn.preprocessing.robustscaler, 2017, [cited 2018 May, 5th]; Available from: http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html.

[26] TensorFlow, 2017, Develop. Available from: https://www.tensorflow.org/tutorials/image_retraining.

[27] G. Bradski, The openCV library, Dr Dobb's J. Softw. Tools (2000).

[28] A. Maurya, 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.