TALLINN UNIVERSITY OF TECHNOLOGY

School of Information Technology

Department of Software Science

MD ASHEK MAHMUD 144813 IVSM

Automatic Detection of Bladderwrack in Underwater Video Stream

Master's Thesis

Supervisor: Juhan-Peep Ernits,

PhD

Co-supervisor: Georg Martin,

PhD

Tallinn
2017

TALLINNA TEHNIKAÜLIKOOL

Infotehnoloogia teaduskond

Tarkvarateaduse instituut

MD ASHEK MAHMUD 144813 IVSM

Põisadru automaatne tuvastamine vee all salvestatud videostriimis

Magistritöö

Juhendaja: Juhan-Peep Ernits,

PhD

Kaasjuhendaja: Georg Martin,

PhD

Tallinn
2017

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Md Ashek Mahmud

15.05.2017

Abstract

The ecosystem of the Baltic Sea is constantly changing. Marine biologists are monitoring various indicators to assess the health of the ecosystem. One of the indicators of the ecosystem health is the spread of Bladderwrack.

Detection of Bladderwrack (*Fucus vesiculosus* L.) is to date performed by humans. In the current research we explore how the detection could be automated. Due to varying visibility conditions under the water, the task is challenging. An existence of a reasonable performing automatic Bladderwrack detection approach would enable to efficiently use an autonomous underwater vehicle for such monitoring tasks. The author of the current thesis processes an underwater recognition framework that uses a fully supervised learning technique with an error resilient classifier.

Bladderwrack classification involves two vital issues: suitable feature representation and optimum classification methodology. Here author analysis the performance of different feature descriptors in terms of accuracy and object detection time by them.

These results suggest that a fast feature descriptor like Speeded Up Robust Feature (SURF), Histograms of Oriented Gradient (HOG) and Fast Retina Keypoint (FREAK) could be used for quick feature extraction and Support Vector Machine (SVM) could be used as classifier to get acceptable performance for Bladderwrack detection in underwater video.

This thesis paper is written in English and has 34 pages, including 5 chapters, 15 figures, and 5 tables.

Annotatsioon

Põisadru automaatne tuvastamine vee all salvestatud videostriimis

Läänemere ökosüsteem on pidevas muutuses. Merebioloogid mõõdavad mere ökoloogilist olukorda erinevate indikaatorite kaudu. Üks selliseid indikaatoreid on põisadru levik.

Põisadru leviku hindamist on merebioloogid seni teinud kas vahetute allveevaatluste või vee all salvestatud video läbivaatuse ja käsitsi annoteerimise kaudu. Käesolevas töös uurime, kuidas seda saaks automatiseerida. Kuna vee all on nähtavus tihti piiratud, siis on ülesanne keerukas. Samas, kui oleks olemas mõistlikult toimiv põisadru tuvastamise lahendus, saaks seda kasutada allveeroboti juhtimiseks põisadru leviku monitoorimise automatiseerimisel. Käesoleva magistritöö autor pakub välja põisadru videost tuvastamise lähenemise, mis kasutab näidetest õppimist veakindla klassifikaatoriga.

Need tulemused näitavad, et kasutades kiiret kujukirjeldajat, näiteks Speeded-Up Robust Feature (SURF), Histogram of Oriented Gradients (HOG) ja Fast Retina Keypoint (FREAK), objekti kujuinformatsiooni kätte saamiseks ja Support Vector Machine (SVM) klassifikaatorina, on võimalik saavutada vastuvõetav kiirus põisadru tuvastamiseks vee all filmitavas videos.

Põisadru klassifitseerimisel on kaks peamist takistust: videost atribuutidena väljendamine ning optimaalse klassifitseerimismeetodi valimine. Siinkohal autor analüüsib erinevate suutlikkust, võrreldes tulemuste täpsust ning kuluvat aega. Tulemused näitavad, et kasutades kiiret kujukirjeldajat näiteks Speeded-Up Robust Feature (SURF), Histogram of Oriented Gradients (HOG) ja Fast Retina Keypoint (FREAK), objekti kujuinformatsiooni kättesaamiseks ja tugivektormasinatel (SVM) põhinevat klassifitseerijat, on võimalik saavutada vastuvõetav kiirus põisadru reaalajas tuvastamiseks vee all filmitavas videos.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 34 leheküljel, 5 peatükki, 15 joonist, ning 5 tabelit.

Table of Contents

List of Figures

List of Tables

List of acronyms

AUV	Autonomous Underwater Vehicle
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
BoW	Bag of Words
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Feature
HOG	Histograms of Oriented Gradient
SVM	Support Vector Machine
KSVM	Kernel Support Vector Machine
NN	Neural Network
FREAK	Fast Retina Keypoint
IDSC	Inner-Distance Shape Context
RBF	Radial Basis Function
HIK	Histogram Intersection Kernel
TP	True Positives
TN	True Negatives
FP	False Positives
FN	False negatives
TRP	True Positive Rate
FPR	False Positive Rate
ms	Milliseconds

1 INTRODUCTION

In recent years many marine biologists have become increasingly interested in underwater video monitoring systems for different underwater flora and fauna surveillance. They have the interest to , for example, detect fish species caught, plankton classification [1], or certain plants like Bladderwrack (*Fucus vesiculosus* L.).. Robotic engineers are trying to develop more enhanced Autonomous Underwater Vehicle (AUV) to detect underwater objects and to localize and measure the distance of the object to provide marine biologists and other interested parties data for the research.

Real-time object detection or classification for robots has been a challenging issue for a long time. It is a great challenging to provide a reliable model for detecting an object in underwater environment for Autonomous Underwater Vehicle (AUV). There have been many good efforts carried out by many researchers to detect objects with Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV) as these typically operate in environments where image quality is good. Thus is even more complex for an Autonomous Underwater Vehicle (AUV) because of various environmental effects as it is very complex to get proper images for image processing and classification captured under water.

Bladderwrack object detection under the water is still a complex task in computer vision as it is necessary to detect in the continuously changing background as the camera is moving, the various colors and shapes of the plant itself, and its different scales. In addition the speed, changing lighting condition in different depth of the sea. To overcome these constraints the author analyzed the result by using different local feature detectors including SURF, FREAK and HOG and for getting better classification author used different versions of Support Vector Machine including most industry accepted kernel function including Gaussian, Polynomial, Chi-squared and Histogram Intersection.

There are many researchers are conducting study [2] [3][31][32] for discovering the effects of environmental changes, reproductivity, environmental tolerances of Bladderwrack and other plants in the Baltic Sea where they found many reason of the changes of the ecosystem including overfishing, drastic decline of marine mammals and eutrophication etc. Detecting and classifying Bladderwrack from other species is difficult which encourage to develop a automatic detection system.

The sample videos and initial problem of Bladderwrack detection from video streams were provided by Dr. Georg Martin from the Estonian Marine Institute, University of Tartu.

1.1 Research Goal

The definite research goal of this thesis is to create a model that will classify and detect Bladderwrack in underwater video captured in the Baltic sea. The main focus of this research is to determine comparatively a better way out of the many image processing techniques which are available in the industry to detect Bladderwrack in real-time as it could be used for adaptively controlling the path or autonomous underwater vehicles.

1.2 Unit of Study

The main unit of the study is to detect Bladderwrack in video frames which are captured in the Baltic sea. In the current approach, the key problem is detecting the feature points in Bladderwrack images and distinguishing them from images that do not contain Bladderwrack in the shortest possible time. The aim is to be able to perform the classification at speeds of 5-10 frame-per-second.

The author plans to use the Bag of Words (BoW) [4,5,6,7,8] model which has two phase detecting feature points in the image using Speeded Up Robust Feature (SURF) [9], Histograms of Oriented Gradient (HOG) [10], Fast Retina Keypoint (FREAK) [11] for detecting feature points in the image. The linear SVM [12] and kernel-based SVM [13] have been used as a classifier for detecting an object.

1.3 Research Questions

I. Can we detect Bladderwrack in the video captured in the Baltic sea using traditional 2D cameras?
II. Which image feature tracking algorithm is the best among SURF, FREAK and HOG for underwater image processing?
III. Which classification algorithm is the best among Linear Support Vector Machine and Kernel-based Support Vector Machine (KSMV) for underwater image classification?

IV. Which combination of feature tracking and classifier algorithm is the most effective in terms of accuracy and computation overhead?

1.4 Organization of the Thesis

This thesis is divided into following chapters:

- **Background and Related Work:** The chapter gives an overview of the feature detection approaches and tools available. Review of related research has been described in this chapter as well.
- **Bladderwrack Detection:** This chapter contains the description of background idea.
- **Bladderwrack Detection Experiments:** The results of the different experiment in this thesis are analyzed and presented.
- **Conclusion:** The outcome of this research is described in this chapter.

2 BACKGROUND AND RELATED WORK

2.1 Related Work

Ma. Shiela Angeli C. Marcos, Maricor N. Soriano and Caesar A. Saloma [14] used neural network based approach to classify coral reefs. They successfully recognized three different types of objects which were corals, dead corals and sand/rubble where they got 86.5% success rates with an NN classifier. They used color and texture as input features of the NN classifier which are attributes used by marine scientists. Their color features and texture features were derived from the r-g chromaticity histogram with mean hue-saturation values and Local Binary Pattern (LBP) respectively. LBP is a visual feature descriptor for image classification.

Emmanuel Okafor, Pornntiwa Pawara, Faik Karaaba, Olarik Surinta, Valeriu Codreanu, Lambert Schomaker and Marco Wiering [6] studied and compared different object detection techniques including deep learning using convolutional neural networks (CNNs), AlexNet and GoogleNet, and BoW with different variations. They found that reducing neuron numbers in the last inception layer of GoogleNet with fully connected layers in AlexNet could provide better performance. They found the best performance from HOG-Bow methods.

Dhanashri Mulmule-Shirkhedkar, Ajay R. Dani [15] compared the performance of SURF and FREAK for feature extraction and identifying Indian currency notes where they used the region based model by choosing unique patches for all types currency banknotes to check against each & every image. They realized competitive results between SURF and FREAK in terms of processing time, memory and accuracy. It showed that FREAK is better as processing time is concerned but in the other hand SURF is better for accuracy and matching.

Drew Schmitt, Nicholas McCoy [16] showed the experimental result of object classification and localization using Naive Bayes and Support Vector Machine (SVM) learning algorithms to classify airplanes, cars, motorbikes, and faces where they used a subset of the CalTech-101 image database. They show ed the performances of the different classifiers against the size of the training sets. Their experiment suggested that Naive Bayes algorithm is the best among Naive Bayes, Linear SVM and Non-Linear SVM. They achieved 81 percent accuracy for Naive Bayes whereas it was 76 percent and 74 percent for Non-Linear SVM and Linear SVM, respectively. In addition to this experiment, they realized better results when they experimented on bigger training sets where they used 400 images. In the second experiment, their success rates were 89 percent, 87 percent and 84 percent for Naive Bayes, Non-Linear SVM and Linear SVM, respectively.

Muhammad Imran Malik, Sheraz Ahmed, Marcus Liwicki and Andreas Dengel [17] performed two signature verification experiments using local feature detectors which are SURF and FAST, where they used FREAK for both the cases. They used publicly available 4NSigComp2010 image database which keeps genuine, forged, and disguised signatures. In their experiments, they obtained identical accuracy from both SURF-FREAK and FAST-FREAK approaches where both had 30 percent error rates, but the interesting fact was FAST-FREAK took only 0.6 second where SURF-FREAK took 12 seconds.

Navneet Dalal and Bill Triggs [18] presents one approach of SVM-based classification technique for human detection where they used Histograms of Oriented Gradients as feature detector. In their approach, there are series of operations for training the model including normalizing gamma and color, computing gradients, weight voting into spatial and orientation cells, contrast normalization over overlapping spatial blocks, collecting HOG's over detection window, applying linear SVM and getting results at the end. They achieved 3% better performance by using Gaussian kernel instead of linear SVM.

S. Guzmán, A. Gómez, G. Diez1, D.S. Fernández [19] show the classification approach for car detection in the outdoor environment using datasets of both 5970 positive and negative images. In their experiment, they detect and count cars using HOG features descriptor and Support Vector Machine as a classifier and discovered that HOG is a good descriptor having identification problems reaching 0.9909 of efficiency with 0.0065 standard deviations.

Zhiyong Wang, Bin Lu, Zheru Chi and Dagan Feng [20] conducted some experiments for leaf image classification where they used 4400 images for training sets and 2200 images for test sets to test 20 different classes of images. They used shape context and SIFT descriptors for feature extraction to utilize local feature and global feature together. Later on, they used KNN classifier for the final classification. They conducted six different experiments with the combination of image size, shape context, utilize contour and venation information. The conclusion of the experiments showed that Histogram of Oriented Gradient-Maximum Margin Criterion (HOG-MMC) having 89.40% accuracy is better than Inner-Distance Shape Context (IDSC) whose accuracy was 83.79%. In addition to the comparison between HOG-

MMC and IDSC, they realized the best performance from their proposed contour information and vein patterns based system which gave 91.30 % accuracy.

Huilin Gao, Lihua Dou, Wenjie Chen and Jian Sun [21] proposed a new approach for image classification with BoW models using improved SIFT algorithm based on uniform grid patches and PCA principles. In their experiment, they reduced the image quality to get better running time. They found a different result from their experiments which showed that traditional SIFT gives 73.33% accuracy whereas SIFT-based on patches and SIFT-based on PCA gives 79.1% and 86.67% accuracy, respectively. In addition to the above experiments, they analyzed the kernel function accuracy where they obtain 86.67% and 90.83 % accuracy from radial basis function (RBF) and histogram intersection kernel (HIK) function.

Sameer Khan, Suet-Peng Yong and Jeremiah D. Deng [22] proposed a system where they developed BoW models using modified SIFT with Harris corner embedding Local Binary Pattern texture feature for classifying medical images. They realized competitive accuracy rates from different experiments using the Support Vector Machine classifier. Experiments showed SIFT has 69.2 % accuracy whereas SIFT+Harris Corner, BOVW (SIFT), BOVW(HSIFT) BOVW(HSIFT-LBP) has accuracy of 71.0%, 58.6%, 73.5%, 80.2%, sequentially.

Balasubramanian T, Krishnan S, Mohanakrishnan M, Ramnarayan Rao K, Vinoth Kumar C and Nirmala K [23] studied an assessment on Glaucoma detection where they used the Histogram of Oriented Gradients (HOG) for feature extraction and SVM for classification. In their approach, they used Gabor filter, morphological operation, thresholding to produce processed image to classify later on. In the SVM they used different kernel functions having different polynomial sets. During their experiment, they achieved 83.3% accuracy with 75% sensitivity glaucoma.

Lhi-Jie Liu [24] proposed another new approach for image classification based on Visual Saliency and Bag of Words Model. He has proposed four steps during the phase, which are sequentially extracting training image using dense sampling method, obtaining a center vector of each cluster using a k-means algorithm, computing histogram of visual words and training samples using SVM. He realized 3% to 4 % better accuracy from his proposed method for different sizes of training sets in contrast to the traditional BoW model.

Yahia Said, Mohamed Atri, Rached Tourki [25] used IHOG features descriptor combined with SVM classifier to propose a system for human detection. In their proposed method they used three step processes, which are Image brightness normalization, Gradient computation, and Histograms computation. They classified the training sets using SVM with different kernel functions like Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid. They managed about 9% better accuracy from their proposed method against normal HOG for positive images, but on the other hand, only 2% less accuracy for negative images.

Guangyuan Zhang, Fei Gao, Cong Liu, Wei Liu and Huai Yuan [26] presents a system for detecting and classifying pedestrians using camera sensors where they used optimized HOG features and SVM classifier. In their approach, they generated optimized HOG features by four steps procedures including Gradient Computation, Spatial/Orientation Binning, Gradient direction histogram and Block Normalisation. In their classifier model, the effect of different kernel functions had been analyzed which were Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid.

2.2 Theoretical Background & Tools

2.2.1 Theoretical Background

Underwater Image: There are many applications and researches are ongoing based on underwater images which are the normal photographic image captured by a conventional optical camera or sonar based camera. Though the underwater image is very difficult to classify due to its nature and environmental effects but still useful for many research like underwater fish classification and counting [27], underwater object Recognition [28], coral reef monitoring [29] etc.

Feature Descriptor: In Image processing or machine learning a feature vector is an n-dimensional vector of numerical features which yields some information of an object. A feature descriptor describes the useful information by extracting information from an image or image area and by avoiding irreverent information kept in that image. It generates a feature vector by converting an image of size width x height x 3. There are many feature descriptors that are being used in computer vision technology like SIFT, SURG, FREAK, HOG etc.

Speeded-Up Robust Features Descriptor: Speeded-Up Robust Features (SURF) is a fast feature descriptor algorithm in the object detection field. In the object recognition field, this algorithm is widely accepted and used by the researchers due to its influential attributes such as scale invariance, contrast immutability, lighting invariance, translation immutability and rotation immutability. However, it can recognize an object in an image which is captured under different external additional and natural settings.

This algorithm has four steps [9]:

a) Generating integral image
b) Detecting interest point
c) Assigning descriptor orientation
d) Generating descriptor.

Figure 1 shows the flow of SURF algorithm.

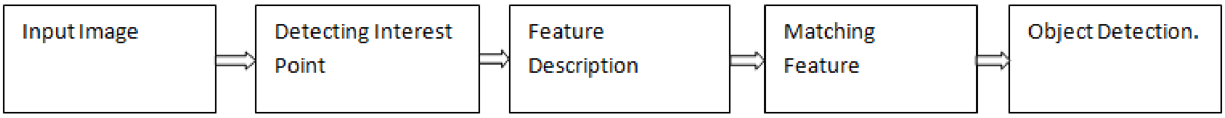


Figure 1: Process flow of SURF algorithm.

Every subsequent step of this algorithm uses an integral image to increase the speed of the process. The integral image is generated by using Eq. (1). When using an integral image, it is necessary to always read only four pixels sto calculate surface integral from the original image.

$$I\sum(x,y) = \sum_{i=0}^{x} \sum_{j=0}^{y} i(i,j) \tag{1}$$

During calculation, Haar wavelet filter and Gaussian fact is used.

$$H_{(x,y)} = det\begin{pmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{pmatrix}$$

$$H(\overline{x}) = D_{xx}(\overline{x})D_{yy}(\overline{x}) - (0.9D_{xy}(\overline{x}))^{2}$$

$$\overline{x} = (x, y, x) \tag{2}$$

This algorithm locates the significant point of the image by using determinants of Hessian matrices which is presented by the Eq. (2) that is modified by the Fast-Hessian detector in two ways. The ways are 1) Convolutions of the image and approximated Gaussian kernels replace Second order partial derivatives by themselves. Box filters with coefficients 1,-1,2,-22 is used for approximation. Eq.(3) shows the coefficient 0:9, which is employed to balance this approximation. 2) The position of the image and their size changes the output of Gaussian kernels.

$$H(x) = H + \frac{\partial H^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^z}{\partial x^z} x$$

$$\hat{x} = \frac{\partial^z H^{-1}}{\partial x^z} \frac{\partial H}{\partial X} \tag{3}$$

The third parameter in the Eq. (3) generates the three-dimensional space of determinant results to describe which scaling differs according to intervals and octaves. All the representative points are considered with same weight by this algorithm which can be achieved by assigning dynamic weight to the representative points. Generally, all the true representative points appear in the training sets but sometimes false selected points appear too. All the representative points carry weight which could be defined by the following formula.

$$W_P = \frac{No.of\ detected\ images\ w.r.t.point\ p}{No.of\ training\ Images\ in\ the\ object} \tag{4}$$

Figure 2: The original image before SURF algorithm is applied.

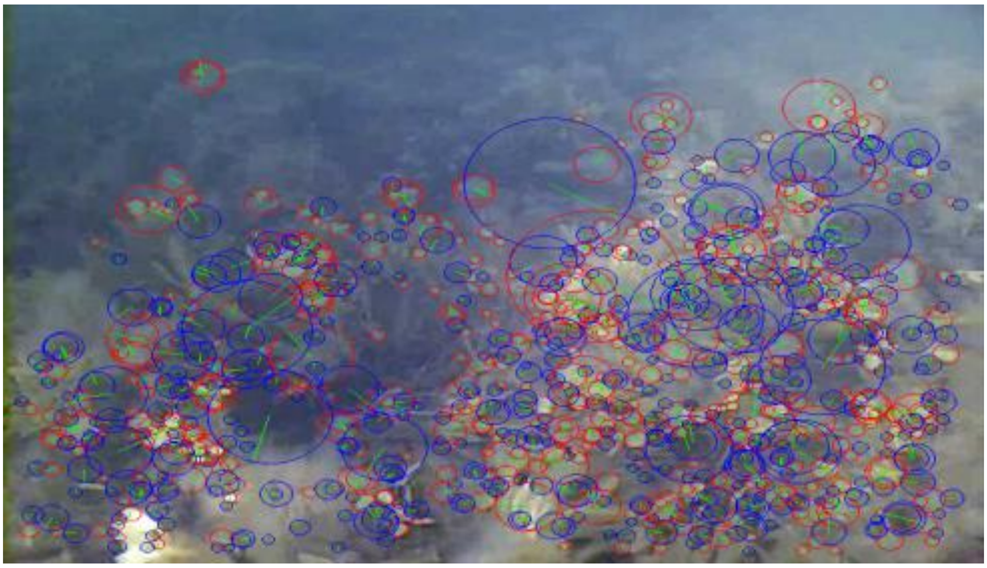


Figure 3: The output of SURF algorithm.

Author has generated the output image in Fig. (3) by applying the SURF algorithm on the original image in Fig. (2). Circles in the image are the detected interest points and the size of circles are scales and the green lines are the orientation where red is the light on the dark and blue is the dark on light.

Fast Retina Keypoint Descriptor: This descriptor is another faster feature points detection algorithm that uses binary descriptor which is generated from the result of brightness comparison test in a number of sampling location around a key point in the image. It detects key points by using corner detectors where AGAST corner detector or Harris corner detector or other detectors could be used.

This algorithm is performed by the following steps [11]:

A. Sampling Pattern

In this step, N number of sample points are located near the given key points and later on curved by Gaussian kernel where the size of the kernel is fixed. Here the sampling points are a receptive field which can be described by the following formula.

$$P_i = P(x_i, y_i) = L_{r_i}(x_i, y_i) \tag{5}$$

where
$$L_{r_i}(x, y) = I(x, y) * G_{r_i}(x, y, \sigma_{r_i}) \tag{6}$$

here, $I\ (x,y)\ represents\ the\ input\ image,\ G_{r_i}(x,y,\sigma_{r_i})$ represents the Gaussian kernel at the point of *i*th receptive filed *where* $L_{r_i}(x,y)$ is the curved version of input image and sampling point P_i is the centre of receptive field r_i is denoted with coordinates $P(x_i,y_i)$.

B. Building the Descriptor

FREAK descriptor is generated upon intensity comparisons between different pair centers of receptive fields which could be defined by the following formula.

$$s(P_a) = f(x) = \begin{cases} 1, & if\ P_i > P_j \\ 0, & else \end{cases}, \tag{7}$$

here $P_a = P(P_i, P_j)$ is considered as a pair of sampling points where $i,j \in \{1,2,3,\ldots\ldots N\}\ and\ i \neq j$. There is a binary encoded intensity comparison s(Pa) applied by this algorithm here.

The comparison constructs the FREAK descriptor F as follow

$$F = \sum_{0 \leq a < N} 2^a s(P_a) \tag{8}$$

C. Orientation normalization

This step calculate the orientation of the orientation of the FREAK descriptor from the pairs of sampling point which are arranged symmetrically with receptive field. The following formula could be used for generating orientation *o*

$$o = \frac{1}{M} \sum_{P_i P_j \in G, i \neq j} (P_i - P_j)\frac{T(P_i) - T(P_j)}{||T(P_i) - T(P_j)||} \tag{9}$$

where, *G* is set of selected pairs, M denotes the number of pair, $T(P_i)$ functions returns 2D coordinates vector at the sampling point *k.*

D. Descriptor matching

In this matching step, matching approach starts with first 128 bits of FREAK descriptors to compare the generated Hamming distance with a predefined threshold, where it eliminates the descriptor if Hamming distance is smaller.

The blue colored small circles in the Fig. 4 represent the selected FREAK descriptors points.

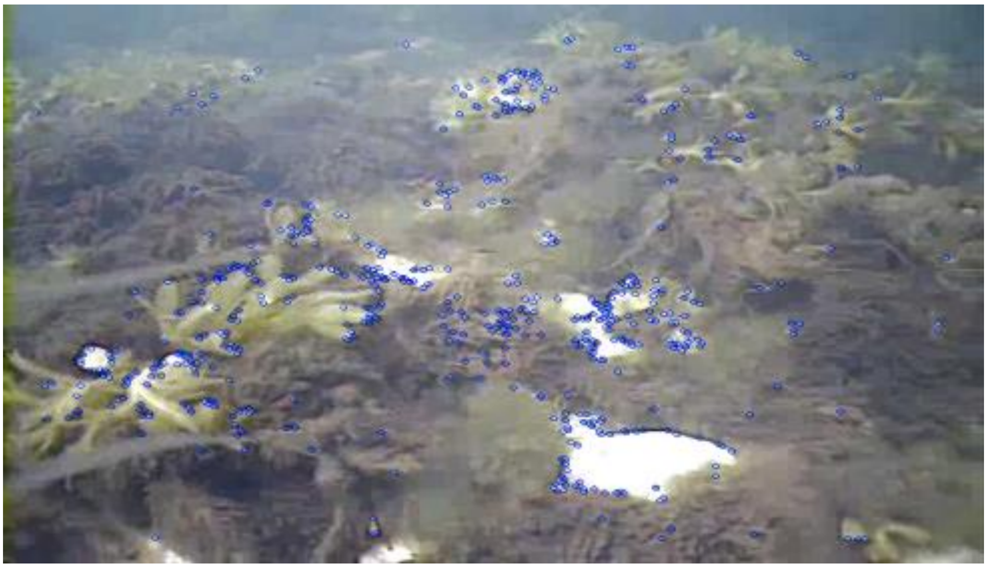


Figure 4: The output of FREAK algorithm

E. Histogram of Oriented Gradients (HOG) descriptor:

The is another feature descriptor algorithm in object detection and recognition technology. It has some similar attributes with SIFT, shape context and cell oriented histograms method. It is generated on a grid of connected cells where it performs overlapping local normalizations to get better performance. This method uses the distribution of intensity gradients or edge directions to represent the shape and appearance of an object located in the image. This algorithm splits the whole image into smaller connected cells and produces a histogram for each cell. The HOG descriptor is represented by the combination of the histograms generated for each cell. This is one of the faster feature descriptors in image processing industry.

This algorithm performs the whole operations by the listed steps [10]:

(i) Gradient Computation

In this first step, it uses some filter mask to calculate the derivative in vertical and horizontal directions. Sobel masks and 1-D (eq. 1) centered kernel filter masks are widely used.

$$D_x = [-1,0,1] \; and \; D_y = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix} \tag{10}$$

(ii) Spatial and Orientation Binning

Here it is considered that I(x,y) is the image intensity at the position of (x,y) where x, y both are derivatives. The gradients of the image in both directions could be calculated by the following convolution operation.

$$\mathrm{I}_x = I * D_x \qquad \text{and} \qquad \mathrm{I_y} = \mathrm{I} * \mathrm{D_y} \tag{11}$$

$$|G| = \sqrt{\mathrm{I}_{x^2} + \mathrm{I}_{y^2}} \qquad \text{and} \qquad \theta = \arctan\frac{\mathrm{I_x}}{\mathrm{I_y}} \tag{12}$$

(iii) Normalization and Descriptor Blocks

In this step, the image is split into a number of cells and a combined histogram of gradient directions generated for each cell, which is calculated for the pixels within their cells. A weighted vote for edge-oriented histogram channel for every pixel is generated by using values from previous steps. Orientations bin, which is generated by combining votes later on spread over 0 to 180 degree.

G. Support Vector Machine

Support Vector Machine [12] was first implemented by Vladimir Vapnik in 1979. An SVM is a supervised learning model that analyze object for classification and regression analysis. Linear SVM separates a set of positive objects from a set of negative objects with maximum margin (Fig. 5), which is defined by the shortest distance between the hyperplane and positive/negative objects.

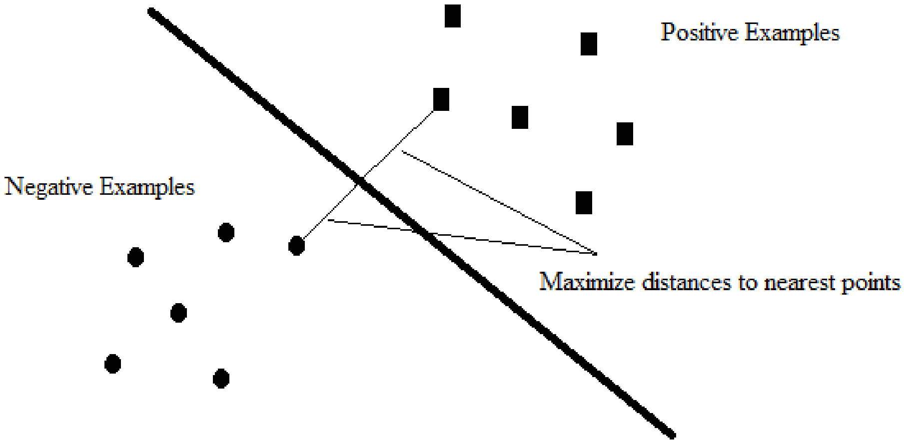


Figure 5: A linear Support Vector Machine

The output of the linear SVM is defined by the flowing formula (Eq. 13) where input vector is x and vector to the hyperplane is w.

$$u = \vec{w}.\vec{x} - b \tag{13}$$

The separating hyperplane can be represented as u=0 . The nearest points to the hyperplane can describe as u = ±1. The margin m could be as following equation.

$$m = \frac{1}{||w||_2} \tag{14}$$

The following formula describes the optimization problem which is occurred while calculating the maximum margin

$$min_{\vec{w},b} \ \frac{1}{2}||\underset{\text{w}}{\rightarrow}||^2 \qquad \text{subject to} \quad y_{i(\vec{w}.\vec{x_i} - b) \geq 1, \forall i} \tag{15}$$

where x_i is the training example, y_i is the SVM output and the value of y_i is +1 or -1 for positive example and negative example respectively.

The Quadratic Programming (QP) problem can be formed by converting optimization problem using a Lagrangian multipliers where N is the number of training examples and Ψ is the objective function that is dependent on the set of Lagrange multipliers αi.

$$\min \psi(\vec{\alpha}) = \underset{\alpha}{min_{\vec{\alpha}}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i\, y_j\, (\overrightarrow{x_i}.\overrightarrow{x_j})\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha \tag{16}$$

In this regards the inequality constraints is

$$\alpha_i \geq o, \forall i \tag{17}$$

and linear equality constraint is

$$\sum_{i=1}^{N} y_i\alpha_i = 0 \tag{18}$$

By using following Lagrange multipliers (Eq. 19) normal vector $\vec{w}$ and the threshold b can be calculated.

$$\vec{w} = \sum_{i=1}^{N} y_i\alpha_i\vec{x}_i \ , b = \vec{w}.\overrightarrow{x_k} - y_k \ for\ some\ \alpha_k > 0 \tag{19}$$

In the case of non-zero support vectors, the computation power needed to evaluate a linear SVM is constant.

I. Kernel Function: It is a function in machine learning algorithm which takes two inputs and splits them based on their similarity. We can get a better classifier from a learning algorithm if we give kernel, images, and labels as input. Kernel function can be denoted as the following formula.

$$K(x_i, x_j) = \emptyset(x_i)^{T}.\emptyset(x_j) \tag{20}$$

There are many types of kernel available in the industry. In this research, I used the following kernel functions.

(i) Gaussian Kernel:

This is one kind of Radial Basis Function (RBF) kernel which could be described by Eq.21.

$$k(x,y) = exp(-\frac{||x-y||^2}{2\sigma^2}) \tag{21}$$

This kernel could be represented by the following Eq.22 too.

$$k(x,y) = exp(-\gamma||x-y||^2) \quad (22)$$

(ii) Polynomial Kernel:

This is a non-stationary kernel and it is appropriate where the normalized training are available. This function can be represented by Eq. 23

$$k(x,y) = (\propto x^T y + c)^{\,d} \quad (23)$$

where $\propto$ is the slope, c is constant and d is polynomial degree.

(iii) Chi-Square Kernel:

This function is constructed from Chi-Square distribution and can be described by the following Eq. 24

$$k(x,y) = 1 - \sum_{i=1}^{n} \frac{(x_i - y_i)^2}{\frac{1}{2}(x_i + y_i)} \quad (24)$$

(iv) Histogram Intersection Kernel:

This well-known image classification kernel which is also known as Min Kernel. It is described as following Eq. 25

$$k(x,y) = \sum_{i=1}^{n} min(x_i, y_i) \quad (25)$$

J. Sequential Minimal Optimization: It [12] is a learning algorithm that solves the quadratic programming (QP) problem in SVM where it does not use any extra memory having no numerical QP optimization steps. It uses Osuna's theorem to decay the SVM-QP problems into OP sub-problems to ensure union between sub-problems. It solves the shortest possible optimization problem at each step. It decides two Lagrange multipliers at each step for joint optimization and calculating optimal values to update the SVM.

This algorithm is selected for the following reasons.

- It works very well on SVM.
- It also performs well for linear SVM because it's computation time is conquered by SVM evaluation.

- It works well with sparse input for both Linear and non-linear SVM because it reduces the kernel computation time.

2.2.2 Relevant Tools

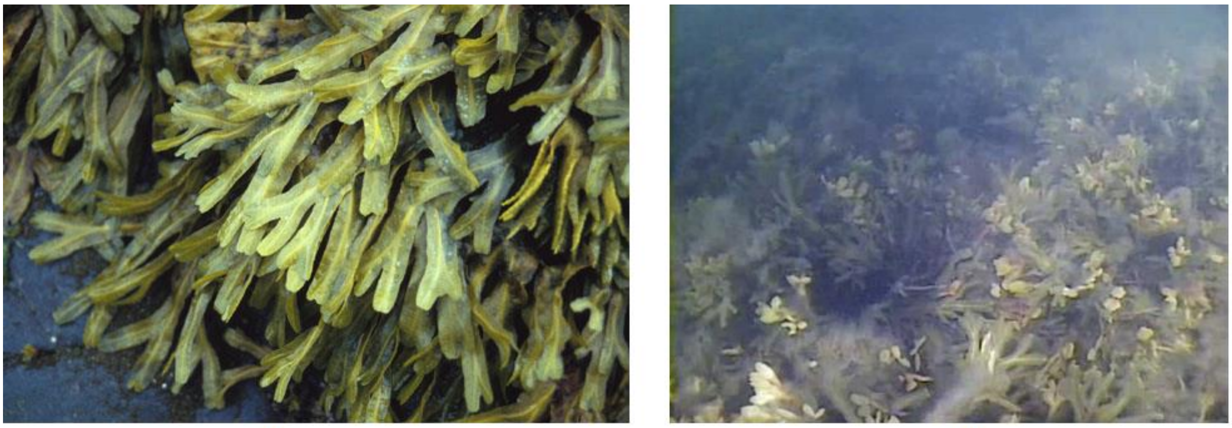
Accord-Framework: It is an open source machine learning framework having audio and image processing libraries. This framework is the extended project of Aforge.net having dedicated libraries for numerical algebra, numerical optimization, artificial neural network and computer vision. This framework uses opencv library as backend. The detail of this framework is available at http://accord-framework.net/.

Aforge.Net: It is an open source framework for computer vision and artificial intelligence, image processing, neural networks, genetic algorithms, fuzzy logic, machine learning, robotics, etc.

Visual Studio 2015: It is a Microsoft provided Integrated Development Environment (IDE) for program development and testing. This tool is chosen because C# is selected as the programming language for experiments since the author is used to with this language.

3. BLADDERWRACK DETECTION

Bladderwrack is a species of brown algae which grow in the tidal zone in North Atlantic and in subtidal zone in the Baltic Sea. The main idea behind this research is finding out the best possible way for real-time classification of the presence of Bladderwrack in the video image which was captured in Baltic Sea by Estonian Marine Institute, University of Tartu. At the beginning of the project, 200 ms was considered as maximum object detection time. Classifying underwater image is very complex due to the low level of image quality for the environmental considerations. We can get better quality Bladderwrack image if we capture outside of the water (Fig 6(a)) whereas it is not clear in (Fig.6 (b)) under the sea level and it becomes worse as long as the deep of the sea level.



(a) (b)

Figure 6: Sample frames containing Bladderwrack.

3.1 Methodology

This research has been conducted in two phases where one is analyzing the performance of the different descriptors. He selected SURF, FREAK and HOG descriptor for feature extraction where author selected these due to their popularity and faster performance. Later on, the author conducted experiments to analyze the performance of different versions of the SVM classier like Linear SVM and Kernel-based SVM with four kernel functions.

3.2 BAG-OF-Words Model

Image information like shape, color, contrast and orientation are the main source of image classification. Author used Bag-of-words model [4,5,6,7,8] which is also known is Bag-of-feature (BoF) or Bag-of-visual feature (BoVF) for the classification process. This model requires extracting features from the learning class images to construct a codebook, later which is used for comparison with extracted feature from unknown image class. The process of this model is presented by the following flowchart (Fig.7).

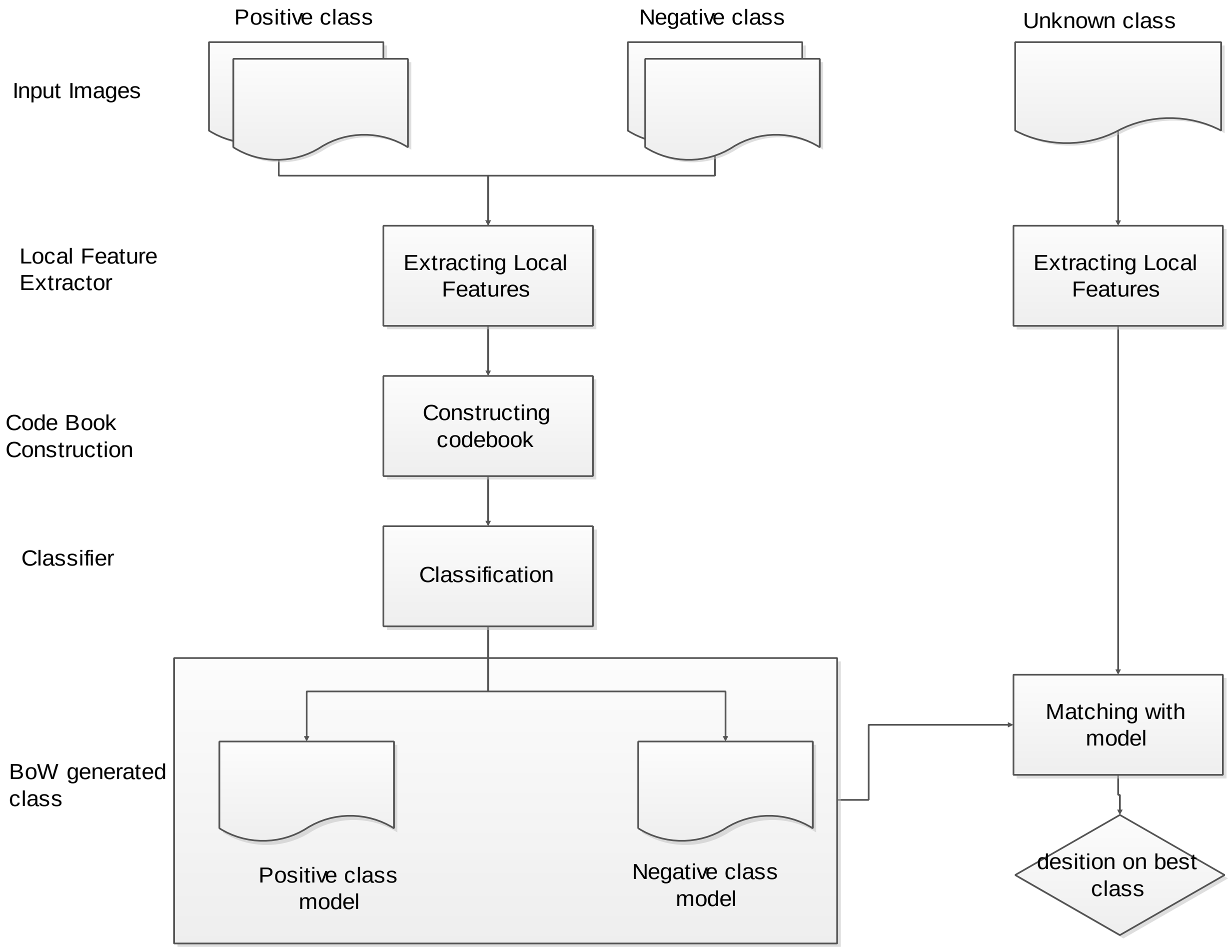


Figure 7: Flowchart of BoW model based image classification.

3.3 Training Procedures

I. At the beginning, training samples are extracted from the image database (see 4.1 Dataset preparation). It loads images as bitmap-image into two dictionaries for the positive and negative image set with their label as Key of the dictionary.

II. This step creates a Bag of Words model or codebook based on the size of the model and the local feature extracted by descriptor where SURF and HOG descriptors use Binary-Split clustering algorithm but FREAK uses K-Modes algorithm for clustering. End of this step feature vector which is an n-dimensional vector that represents the numerical presentation of an image is created from the local features of the images. It is generated by applying the transformation functions to the input. Actually, it generates the value of y from the input vector x.

III. In this step, SVM is created by using Sequential Minimal Optimization algorithm. The details of this SMO (3.4 Learning Procedures of the SVM) is described in next section.

3.4 Learning Procedures of the SVM

For the SMO [30] suppose there is a dataset $(x_1, y_1) \ldots \ldots \ldots . (x_n, x_n)$ where x_i is the input vector and $y_i\{-1, +1\}$ is associated binary label. Support Vector Machine could be trained by the following formula.

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \, y_j K(x_i, x_j) \alpha_i \alpha_j, \tag{26}$$

where $o \leq \propto_i \leq C, \; for \; i = 1,2, \ldots \ldots ., n,$ and

$$\sum_{i=1}^{n} y_i \alpha_i = 0$$

here SVM hyper parameter is C, the kernel function is $K(x_i, x_j)$ and Lagrange multipliers based variables are α_i.

This algorithm breaks down the optimization problem into sub-problems to be solved analytically. The smallest possible problems engages two smallest multipliers since there is linear quality constraint involving the Lagrange multipliers α_i. The constrains are reduced for two multipliers α_i and α_j into the following problem.

$$o \leq \alpha_i, \alpha_2 \leq C$$

$$y_1\alpha_1 + y_2\alpha_2 = k,$$

This reduced problem could be solved analytically by finding a minimum of a one-dimensional quadratic function for one of them. K is the fixed negative in each iteration.

This algorithm is executed as by the following steps.

I. In first step it finds the Lagrange multiplier α_i
II. Then it picks a second multiplier α_2 to optimize the pair.
III. It iterates step 1 and 2 until convergence.

At the end of the iterations, a SVM is created where that uses only the points whose Lagrange multiplier is positive. The expected outputs y_i generates a single vector **w** by multiplying individual associated Lagrange multipliers a_i with them.

$$F(x) = \sum_{i=0}^{N} \{\alpha_i y k(z_i, x)\} + b = \sum_{i=0}^{N} \{w_i k(z_i, x)\} + b \tag{27}$$

This algorithm is also controlled by three parameters beside the kernel function.

a. Complexity (cost) parameter C: Bigger value of C gives more accurate result but that result might not be generalizly well.
b. Sigma **(ε):** This parameter manages the width of the ε-insensitive zone. The bigger ε produces less support vector machine with higher detection rate but less accurate result.
c. Tolerance (T): This is the criterion for completing the training process.

3.5 Detection process

I. At the first step, feature vector is created from the input bitmap-image using feature vector descriptor.

II. The class-level decision against the input feature vector is computed by the appropriate SVM in this final step.

3.6 Confusion Matrix

The performance of the classifier is evaluated during this research in terms of accuracy and average detection time of the image processing for each frame in the video file. Author used confusion matrix to present the performance of different classifier using a predefined set of test images where the true classes are known. Here is a sample confusion matrix for a binary classier given in Table 1.

Table 1: Sample confusion matrix.

n=10	Detected : Yes	Detected : No
Actual: Yes	6	4
Actual: No	4	6

The generalized basic terms used in this matrix are listed blew.

a) True positives (TP): In that case where both actual case and system decided case both are YES.

b) True negatives (TN): In that case where both actual case and system decided case both are NO.

c) False positives (FP): In that case where the actual case is NO but system decided case is YES.

d) False negatives (FN): In that case where the actual case is YES but system decided case is NO.

This confusion matrix can give us the following information.

System's accuracy:

$$\text{Accuracy} = (\text{True positives (TP)} + \text{True negatives (TN)})/n \times 100\% \qquad (26)$$

System's Error Rate:

$$\text{Error Rate} = (\text{False positives (FP)} + \text{False negatives (FN)})/n \times 100\% \qquad (27)$$

System's True Positive Rate:

$$\text{TPR} = \text{True positives (TP)}/ n \times 100\% \qquad (28)$$

System's False Positive Rate:

$$\text{FPR} = \text{False positives (FP)}/n \times 100\% \qquad (29)$$

System's Precision:

$$\text{Precision} = \text{True positives (TP)}/(\text{decided true}) \times 100\% \qquad (30)$$

4. BLADDERWRACK DETECTION EXPERIMENTS

In this chapter author represents the experiments performed, the result found and the analysis of the results.

4.1 Dataset preparation

Author received video footage from the Estonian Marine Institute, University of Tartu and which processed by converting video to frames of still images to create our experimental dataset. This dataset has 250 Bladderwrack containing positive images and 250 negative images where they are scaled into 350x262 pixels having aspect ratio properly. The following images are the examples of positive image set (Fig. 8) and negative image set (Fig. 9).

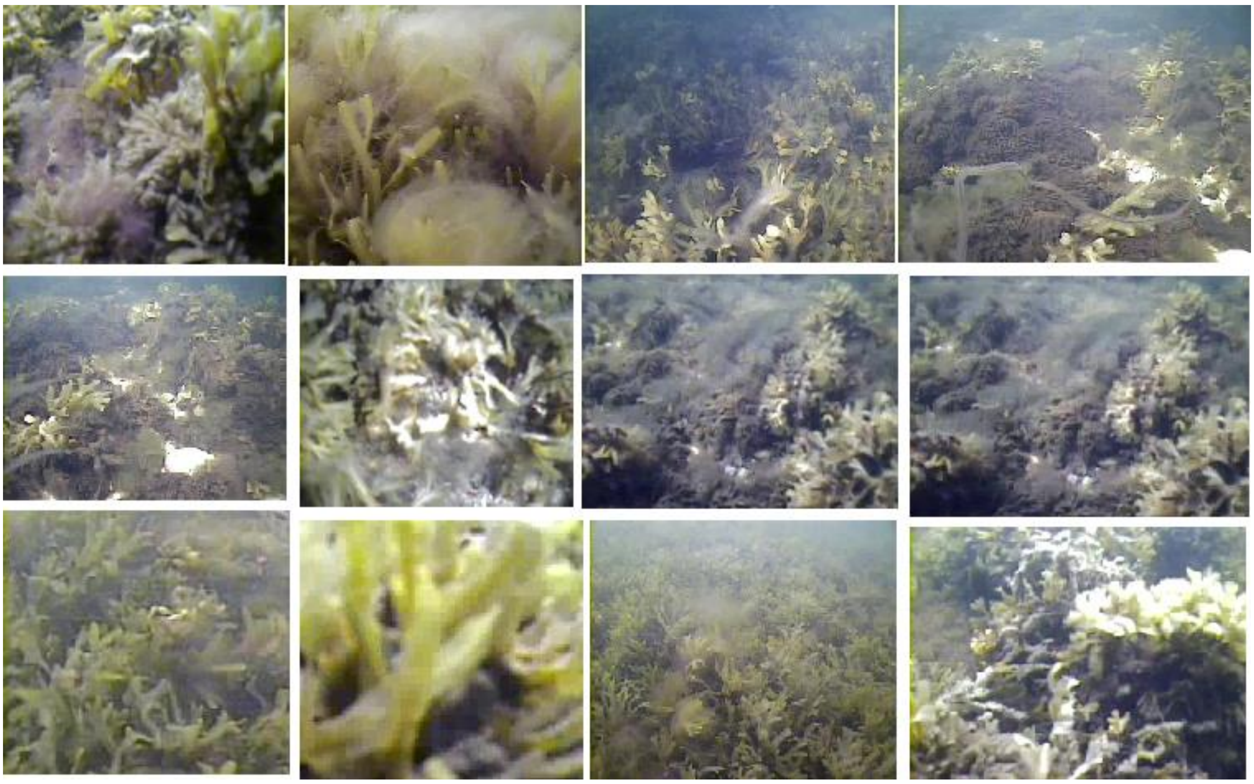


Figure 8: Sample of the positive image set.

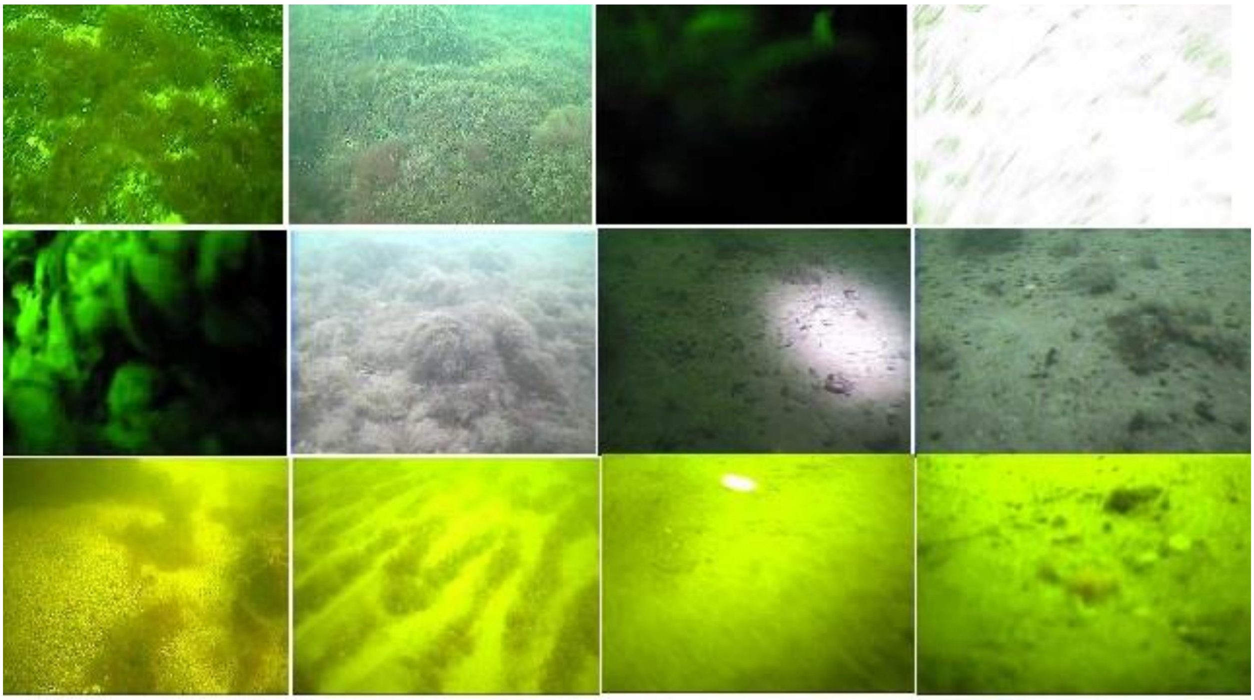


Figure 9: Sample of the negative images set.

4.2 Initial Classification with Support Vector Machine

In the BoW model, choosing the number of visual words is a vital decision because it influences the result of the classifier in terms of accuracy and detection time of the system. The following two subsections represents the result of accuracy and the detection time of the system found from this classifier. The detail of the result of this classifier is listed in Table 2.

4.2.1 Classifier accuracy for different size of visual words

It is clearly visible in Fig. 10 that number of visual words influences the accuracy of the classier sharply. It is shown that the accuracy of the SURF, FREAK and HOG feature descriptor based approach are 78.50%, 79.00%, and 82.00% respectively when the visual words size is 50 whereas it is 87.00%, 81.00%, and 87.00% sequentially when the words size is 100. In addition to this, the accuracy of the classifiers are 90.00%, 80.50%, 83.50% in the order where the words size is 250. Increased number of visual words does not give significantly better performance all the time and it does not change the detection time considerably according to words size which is described in next section.

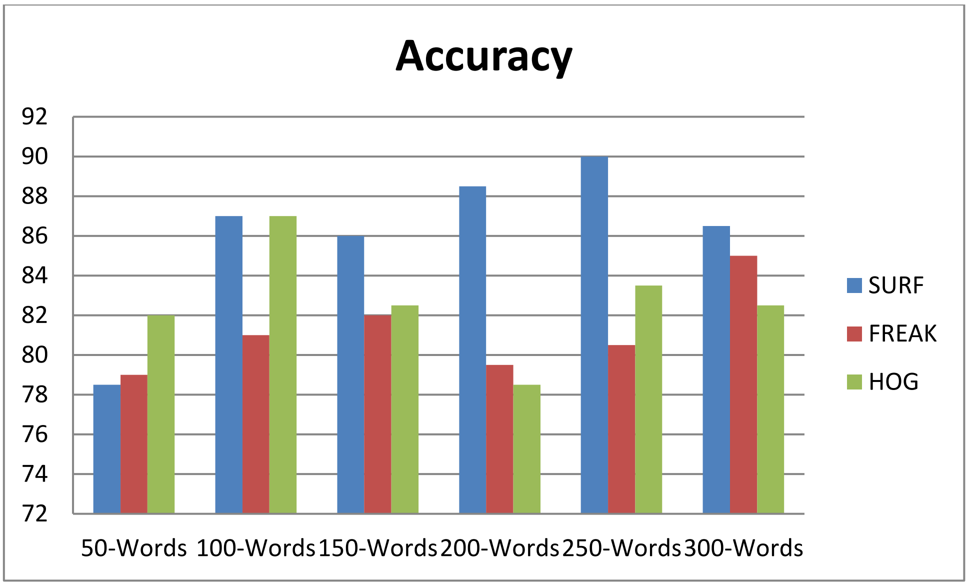


Figure 10: Classifier accuracy for different size of visual words.

4.2.3. Detection time for different size of visual words

The Bladderwrack detection time depends on the feature descriptor and number of visual words used in this model. The detection time for different feature descriptor in different size of visual words is shown in Fig. 11. It is clearly presented that SURF descriptor-based approach takes too much extra time than other descriptors. It takes 132 milliseconds where the size of the visual words is 50 whereas FREAK and HOG takes 23 ms and 17ms respectively. The detection time of SURF, FREAK and HOG descriptor are 105ms, 30ms, and 32ms sequentially where the size of codebook is 300. The detection time is almost stable in all cases where the size of the codebook are 50,100,150,200,250 and 300. The initially targeted detection time was 200 ms which could be possible to achieve by using any of the feature descriptors. The detection time with other associated parameters of experiments is listed in Table 2.

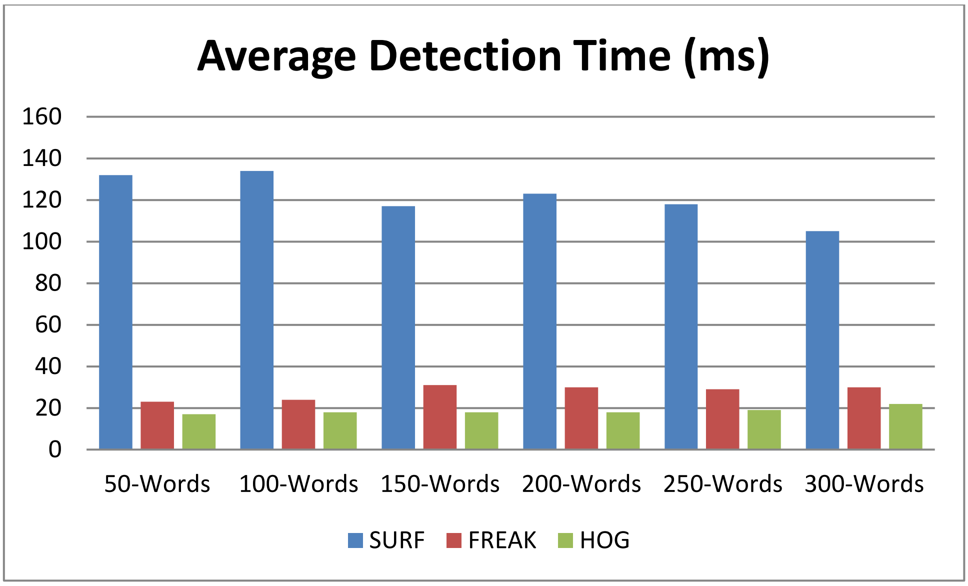


Figure 11: Average detection time for the different size of visual words.

4.2.4. The result of classification experiment based on the size of visual words.

The result of different classifications based on different size visual words and different feature descriptor is listed here. The best performance of FREAK is 94% accuracy when the visual words size is 300 whereas and the best result of SURF and HOG are 93% and 92% sequentially where their visual words size is 250.

Table 2: Result of classification experiment based on the size of visual words.

Serial No	Feature Descriptor	No of Words	Average Detection Time (ms)	True Positive (%)	False Positive (%)	True Negative (%)	False Negative (%)	Accuracy (%)	Error Rate (%)
1	SURF	50	132	82	18	75	25	78.50	21.50
2	FREAK	50	23	80	20	78	22	79.00	21.00
3	HOG	50	17	89	11	75	25	82.00	18.00
4	SURF	100	134	93	7	81	19	87.00	13.00
5	FREAK	100	24	91	9	71	29	81.00	19.50

6	HOG	100	18	95	5	79	21	87.00	13.00
7	SURF	150	117	94	6	78	22	86.00	14.00
8	FREAK	150	31	89	11	75	25	82.00	18.00
9	HOG	150	18	85	15	80	20	82.50	17.50
10	SURF	200	123	93	7	84	16	88.50	11.50
11	FREAK	200	30	87	13	72	28	79.50	20.50
12	HOG	200	18	87	13	70	30	78.50	21.50
13	***SURF***	***250***	***118***	***93***	***7***	***87***	***13***	***90.00***	***10.00***
14	FREAK	250	29	83	17	78	22	80.50	19.50
15	***HOG***	***250***	***19***	***92***	***8***	***75***	***25***	***83.50***	***16.50***
16	SURF	300	105	89	11	84	16	86.50	13.00
17	***FREAK***	***300***	***30***	***94***	***6***	***76***	***24***	***85.00***	***15.00***
18	HOG	300	22	88	12	77	23	82.50	17.50

4.3. Classification with kernel-based Support Vector Machine

The SVM classifier gave acceptable accuracy when the size of the visual words is 300 which is the baseline for additional experiments with different kernel functions in this phase. There are many kernel functions available in the industry. The kernel functions were chosen for additional experiments are Gaussian, Polynomial, Chi-squared and Histogram Intersection kernel because of their higher rate of acceptance in the industry. The detail results of the kernel based SVM classifier are listed in Table 3,4 and 5.

4.3.1. Classifier accuracy for different kernel functions

Kernel functions can control the performance of the classifier. Hence, the kernel-based SVM with SURF, FREAK and HOG feature descriptors are used for experiments and the summary of those experiments is presented in Fig.12. The result shows that Gaussian kernel gives the same result for each of the descriptors which are in between 51.00% and 54.00% of accuracy. The chi-squared and Histogram Intersection kernel give a steady performance for the different descriptor in between 80.00-90.00 % of accuracy. In addition to this, the Polynomial kernel gives interesting fact where it represents that the accuracy of FREAK is very low which is 61.00% whereas the performance of SURF and HOG is 86.50% and 81.50 % respectively.

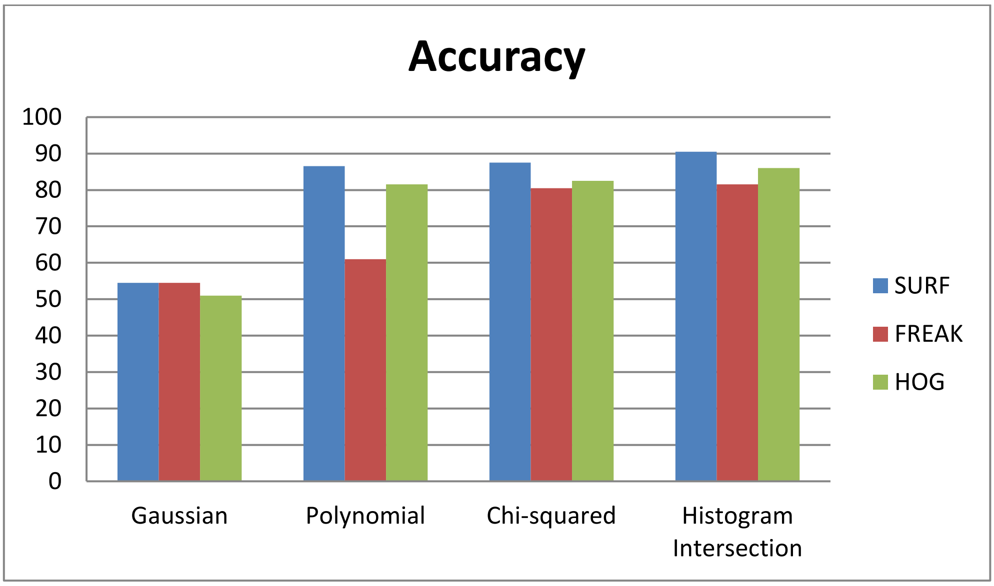


Figure 12: Classifier accuracy for different kernel functions.

4.3.2. Detection time for different kernel functions

The effect of different size of visual words has been discuss used in earlier section 4.2.3 and the how the different kernel functions manipulate the detection time is presented here. In this experiment size of the visual words is 300 and fixed for each descriptor and kernel functions. The summary of the experiments is simply presented by Fig. 13 where it is clearly visible that all types of the combination of feature descriptor and kernel functions give the steady detection time. The FREAK and HOG descriptor take very less time than SURF descriptor where whatever the kernel function they use. They take less than 40 ms for any kernel but the SURF takes more than 120ms for any kernel.

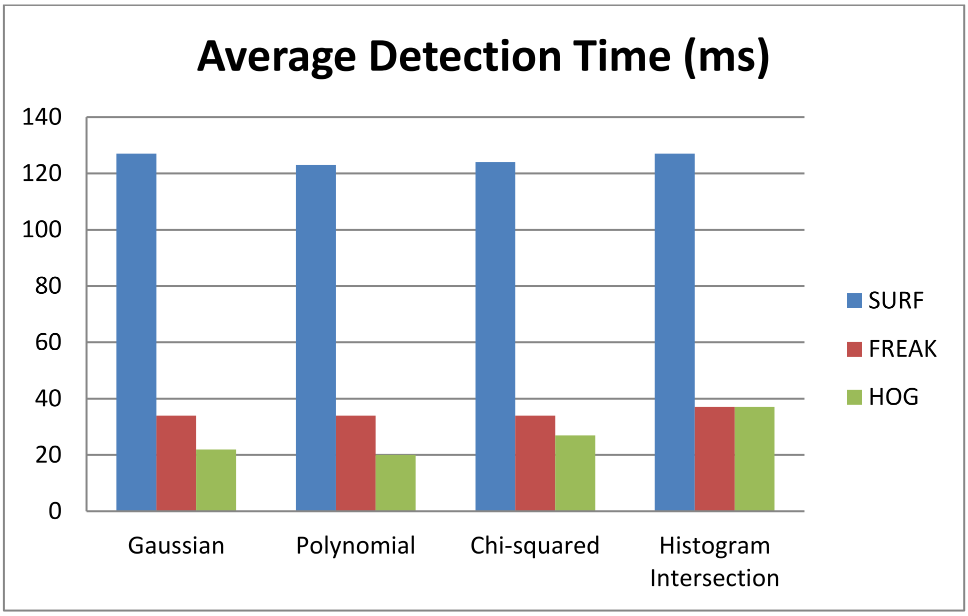


Figure 13: Average detection time for kernel functions.

4.3.3. The result of classification experiment based on kernel function.

A. Kernel function experiment based on SURF detector

Histogram Intersection kernel provides best performance as 90.5% accuracy.

Table 3: Result of kernel function experiment based on SURF detector.

Serial No	Kernel	Average Detection Time (ms)	True Positive (%)	False Positive (%)	True Negative (%)	False Negative (%)	Accuracy (%)	Error Rate (%)	Precision (%)
1	Gaussian	127	100	0	9	91	54.5	45.5	52.36
2	Polynomial	123	90	10	83	17	86.5	13.5	58
3	Chi-squared	124	95	5	80	20	87.5	12.5	79.61
4	***Histogram Intersection***	***127***	***98***	***2***	***83***	***17***	***90.5***	***9.5***	***78.9***

B. Kernel function experiment based on FREAK detector.

Histogram Intersection kernel provides best performance as 81.5% accuracy.

Table 4: Result of kernel function experiment based on FREAK detector.

Serial No	Kernel	Average Detection Time (ms)	True Positive (%)	False Positive (%)	True Negative (%)	False Negative (%)	Accuracy (%)	Error Rate (%)	Precision (%)
1	Gaussian	34	100	0	9	91	54.5	45.5	52.36
2	Polynomial	34	96	4	80	20	61	39	84.11
3	Chi-squared	34	96	4	79	21	80.5	19.5	82.61
4	***Histogram Intersection***	***37***	***100***	***0***	***77***	***23***	***81.5***	***18.5***	***85.22***

C. Kernel function experiment based on HOG detector.

Histogram Intersection kernel provides best performance as 86% accuracy.

Table 5: Result of kernel function experiment based on HOG detector.

Serial No	Kernel	Average Detection Time (ms)	True Positive (%)	False Positive (%)	True Negative (%)	False Negative (%)	Accuracy (%)	Error Rate (%)	Precision (%)
1	Gaussian	22	100	0	2	98	51	49	50.51
2	Polynomial	20	96	4	67	33	81.5	18.5	74.42
3	Chi-squared	27	96	4	69	31	82.5	17.5	75.59
4	***Histogram Intersection***	***37***	***96***	***4***	***76***	***4***	***86***	***14***	***80***

4.4 Final System Output

This section represents the final output of the system. The classification true positive result in Fig. 14 and true negative result in Fig. 15 have been shown here. This system processes each frame in the video in real time mode and calculate the classification result to display the result on the frame. This system does not show 100% accurate result as it is shown that none of the classification gives fully correct result and some frame in the video has less quality of image than the training set.

Figure 14: Final system output (Positive).

Figure 15: Final system output (Negative).

4.5 Summary of the experiments

At the beginning of the research, the author set the acceptable accuracy as 80%. All the experimental results are listed in the previous section of this chapter. According to the results, It can be said that the number of visual words should be 300 because each of the feature descriptors gives at least 80% accuracy with Linear SVM. In addition to this, it keeps the detection time in the acceptable range but it is very much clear that SURF detector is very slower than FREAK and HOG.

The effect of different kernel functions have been experimented in this research as well and it gives some significant variation in classification result. The SURF descriptor gives the best accuracy when it works with HI kernel but it takes huge computation time. In the other hand, the performance of FREAK and HOG is almost close to SURF but take very low computation time. Hence it could be concluded that FREAK or HOG descriptor with Chi-squared or HI kernel could be the best combination for Bladderwrack detection.

Our one of the initial goal of this research was keeping the detection lower than 200 ms so that it can process at least 5 frames per second. It can be summarized that system has achieved the both goals.

5. CONCLUSION

Bladderwrack detection is a new type of object detection system in the context of image detection recognition fields. Marine biologists have interest in monitoring Bladderwrack spread in the sea which is very much complex to them due to environmental limitation. The approach can be used to adapt missions of Autonomous Underwater Vehicles in real time.

There are many technologies and methodologies are being used and implemented for the purpose of underwater object detection in order to meet the user requirements.

In this thesis, author proposes a method which could be one of the potential approaches to build Bladderwrack detecting system. The presented approach is fast enough to be performed several images per second on a commodity CPU. Thus it is possible to avoid expensive tools like high regulation camera and processing the input from another sensor at the time of carrying out a mission. This proposed system works with normal grade video captured by general underwater video camera.

Author used a provided set of video footage to generate images from video frame to build training dataset. Author manually labeled images into two groups, positive group and negative group in order to train the model using SVM classifier. The initial goals of this research were achieving at least 80 % accuracy and maintaining detection time less than 200 ms to process 5 frames per second. The author have achieved the goals in terms of both accuracy and detection time.

During this research, the author had only a limited amount of video footage captured in Baltic Sea that's why author developed the system based on that restricted environment and tested the system based on these video images only. It could be difficult to confirm the system performance in another environment as it is not testeded in other environments. Author believe this research could be a good starting for further research on Bladderwrack detection and recognition in the different angle.

REFERENCES

1 Sander Dieleman. 2015. Classifying plankton with deep neural networks. (MARCH 17, 2015). Retrieved April 2, 2017 from http://benanne.github.io/2015/03/17/plankton.html

2 Kautskyl Nils, Kautskyl Hans, Kautskyl Ulrik and Waern Mats. 1985. Decreased depth penetration of Bladderwrack vesiculosus (L.) since the 1940's indicates eutrophication of the Baltic Sea. Asko laboratory. Institute of Marine Ecology and the Department of Zoology, University of Stockholm, S-106 91 Stockholm, Sweden, Institute of Plant Ecology. University of Uppsala. Uppsala. Sweden. Vol.28. 1-8. October 1985

3 Kolesova N., Kõuts M., Siimon K. L. and Raudsepp U. 2013. Changes in the morphology of Bladderwrack vesiculosus L. and abundance of seaweed associated fauna along the coastal sea of Estonia. Marine Systems Institute at Tallinn University of Technology.

4 Xu Sheng, Fang Tao, Li Deren, and Wang Shiwei. 2010. Object Classification of Aerial Images With Bag-of-Visual Words. IEEE Geoscience and Remote Sensing Letters. Vol. 7, No. 2. 366-370. April 2010. doi: 10.1109/LGRS.2009.2035644

5 Zare M. Reza , Ahmed Mueen and Seng W. Chaw. 2013. Automatic classification of medical X-ray images using a bag of visual words. IET Computer Vision. Vol. 7, No. 2. 105-114. April 2013. doi: 10.1049/iet-cvi.2012.0291.

6 Melloni A., Bestagini P., Costanzo A. and Barni M. 2013. Tagliasacchi M. and Tubaro S. Attacking image classification based on bag-of-visual-words. 2013 IEEE International Workshop on Information Forensics and Security (WIFS). Guangzhou, 2013. 103-108. doi: 10.1109/WIFS.2013.6707802.

7 Ali N. Mohd, Jun S. Wei,Karis M. Safirin,Ghazaly M. Md, and Aras M. S. Mohd. 2016. Object classification and recognition using Bag-of-Words (BoW) model. 2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA). Malacca City,

2016. 216-220. doi: 10.1109/CSPA.2016.7515834

8 Okafor Emmanuel, Pawara Pornntiwa, Karaaba Faik, Surinta Olarik, Codreanu Valeriu, Schomaker Lambert and Wiering Marco. 2016. Comparative study between deep learning and bag of visual words for wild-animal recognition. 2016 IEEE Symposium Series on Computational Intelligence (SSCI). Athens, 2016. 1-8. doi: 10.1109/SSCI.2016.7850111

9 Herbert Bay, Tinne Tuytelaars and Gool L. Van. 2008. SURF: Speeded Up Robust Features. In Computer Vision and Image Understanding (CVIU) (2008). 346-359.

10 Ali K. Jaber and Ikhlas Abdel-Qader. 2016. Hybrid Histograms of Oriented Gradients-compressive sensing framework feature extraction for face recognition. 2016 IEEE International Conference on Electro Information Technology (EIT). Grand Forks, ND, 2016. 0442-0447. doi: 10.1109/EIT.2016.7535281

11 Yanhai Wu, Cheng Zhang, Jing Wang and Nan Wu. 2015. Image registration method based on SURF and FREAK. 2015 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC). Ningbo, 2015. 1-4. doi: 10.1109/ICSPCC.2015.7338825

12 John C. Platt. 1998. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. Technical Report MSR-TR-98-14. Microsoft Research. April 21, 1998

13 César Souza. 2015. Kernel Functions for Machine Learning Applications. MARCH 17, 2010. Retrieved April 5, 2017 from http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

14 Ma. Shiela Angeli C. Marcos, Maricor N. Soriano and Caesar A. Saloma. 2005. Classification of coral reef images from underwater video using neural networks. Optics Express. 8766-8771. October 2005

15 Dhanashri Mulmule-Shirkhedkar and Ajoy R. Dani. 2015. Comparative study of SURF and FREAK descriptor on Indian Rupee Currency notes. 2015 International Conference on Information Processing (ICIP). Pune, 2015. 784-789. doi: 10.1109/INFOP.2015.7489488

16 Schmitt Drew and McCoy Nicholas. 2011. Object Classification and Localization Using SURF Descriptors. December 13, 2011. Retrieved April 5, 2017 from http://cs229.stanford.edu/proj2011/SchmittMcCoy-ObjectClassificationAndLocalizationUsingSURFDescriptors.pdf

17 Malik M. Imran, Ahmed Sheraz, Liwicki Marcus and Dengel Andress. 2013. FREAK for Real Time Forensic Signature Verification. 2013 12th International Conference on Document Analysis and Recognition. Washington, DC. 2013. 971-975. doi: 10.1109/ICDAR.2013.196

18 Dalal Navneet and Triggs Bill. 2005. Histograms of oriented gradients for human detection. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). San Diego, CA, USA, 2005. 886-893 Vol. 1. doi: 10.1109/CVPR.2005.177

19 Guzman Sebastian, Gomez Alexander, Diezv German and Fernandez David. 2015. Car detection methodology in outdoor environment based on histogram of oriented gradient (HOG) and support vector machine (SVM). 6th Latin-American Conference on Networked and Electronic Media (LACNEM 2015). Medellin, 2015. 1-4. doi: 10.1049/ic.2015.0310

20 Wang Zhiyong, Lu Bin, Chi Zheru and Feng Dagan. 2011. Leaf Image Classification with Shape Context and SIFT Descriptors. 2011 International Conference on Digital Image Computing: Techniques and Applications. Noosa, QLD, 2011. 650-654. doi: 10.1109/DICTA.2011.115

21 Gao Huilin, Dou Lihua, Chen Wenjie and Sun Jian. 2013. Image classification with Bag-of-Words model based on improved SIFT algorithm. 2013 9th Asian Control Conference

(ASCC). Istanbul, 2013. 1-6. doi: 10.1109/ASCC.2013.6606268

22 Khan Sameer, Yong Suet-Peng and Jeremiah D. Deng. 2015. Ensemble classification with modified SIFT descriptor for medical image modality. 2015 International Conference on Image and Vision Computing New Zealand (IVCNZ). Auckland, 2015. 1-6 doi: 10.1109/IVCNZ.2015.7761517

23 Balasubramanian T., Krishnan S., Mohanakrishnan M., Ramnarayan Rao K., Vinoth Kumar C. and Nirmala K. 2016. HOG feature based SVM classification of glaucomatous fundus image with extraction of blood vessels. 2016 IEEE Annual India Conference (INDICON). Bangalore, 2016. 1-4. doi: 10.1109/INDICON.2016.7838902

24 Zhi-Jie Liu. 2015. Image Classification Method Based on Visual Saliency and Bag of Words Model. 2015. 8th International Conference on Intelligent Computation Technology and Automation (ICICTA). Nanchang, 2015, 466-469. doi: 10.1109/ICICTA.2015.122

25 Said Yahia, Atri Mohamed and Tourki Rached. 2011. Human detection based on integral Histograms of Oriented Gradients and SVM. 2011. International Conference on Communications, Computing and Control Applications (CCCA). Hammamet, 2011. 1-5. doi: 10.1109/CCCA.2011.6031422

26 Zhang Guangyuan, Gao Fei, Liu Cong, Liu Wei and Yuan Huai. 2010. A pedestrian detection method based on SVM classifier and optimized Histograms of Oriented Gradients feature. 2010 Sixth International Conference on Natural Computation. Yantai, Shandong, 2010. 3257-3260. doi: 10.1109/ICNC.2010.5582537

27 Kim T. Won, Yu Son-Cheol and Yuh Junku. 2008. Neural network-based underwater image classification for Autonomous Underwater Vehicles. 17th World Congress The International Federation of Automatic Control Seoul. Seoul, Korea 2008. 15991-15996

28 Chuang Meng-Che, Hwang Jeng-Neng and Williams Kresimir. 2016. A Feature Learning and Object Recognition Framework for Underwater Fish Images. IEEE Transactions on

Image Processing. Vol. 25, No. 4. 1862-1872. April 2016. doi: 10.1109/TIP.2016.2535342

29 Gleason A.C.R, Reid R.P. and Voss K. J. 2007. Automated classification of underwater multispectral imagery for coral reef monitoring. OCEANS 2007. Vancouver, BC, 2007. 1-8. doi: 10.1109/OCEANS.2007.4449394

30 César Souza. 2010. Learning Algorithms. APRIL 27, 2010 . Retrieved April 5, 2017. from http://crsouza.com/2010/03/17/kernel-functions-for-machine-learning-applications/

31 Torn Kaire, Krause-Jensen Dorte,Martin Georg. 2006. Present and past depth distribution of bladderwrack (Fucus vesiculosus) in the Baltic Sea. Aquatic Botany. Vol. 84, Issue 1, January 2006, 53-62, ISSN: 0304-3770. doi:10.1016/j.aquabot.2005.07.011

32 Ojaveer Henn, Jaanus Andres, MacKenzie BR, Martin Georg, Olenin Sergej. 2010. Status of Biodiversity in the Baltic Sea. PLOS ONE 5(9): e12467. doi:10.1371/journal.pone.0012467