Temporal Frame Difference Using Averaging Filter for Maritime Surveillance

Abdullah I. A. Alfadda

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Electrical & Computer Engineering

Dhruv Batra, Chair

A. Lynn Abbott

Devi Parikh

July 20, 2015

Blacksburg, Virginia

Keywords: Maritime, Machine Learning, Image Processing

Temporal Frame Difference Using Averaging Filter for Maritime Surveillance

Abdullah I. A. Alfadda

(ABSTRACT)

Video surveillance is an active research area in Computer Vision and Machine Learning. It received a lot of attention in the last few decades. Maritime surveillance is the act of effective detection/recognition of all maritime activities that have impact on economy, security or the environment. The maritime environment is a dynamic environment. Factors such as constant moving of waves, sun reflection over the sea surface, rapid change in lightning due to the sun reflection over the water surface, movement of clouds and presence of moving objects such as airplanes or birds, makes the maritime environment very challenging. In this work, we propose a method for detecting a motion generated by a maritime vehicle and then identifying the type of this vehicle using classification methods. A new maritime video database was created and tested. Classifying the type of vehicles have been tested by comparing 13 image features, and two SVM solving algorithms. In motion detection part, multiple smoothing filters were tested in order to minimize the false positive rate generated by the water surface movement, the results have been compared to optical flow, a well known method for motion detection.

To my mother & father

Acknowledgments

At first, I would like to thank my advisor Dr. Drhuv Batra, for his advises and time throughout my M.S. program. Also, I would like to thank my committee members Dr. Lynn Abbott and Dr. Devi Parikh for their willingness to be part of my committee. Also, I would like to pay special thanks for Dr. Lynn for reviewing my thesis word by word, and giving a detailed feedback.

Also, I would like to pay special thanks to my sponsor King Abdulaziz City for Science & Technology (KACST) for awarding me the scholarship, and giving me the opportunity to pursue my graduate studies. Also I would like to thank Saudi Arabian Cultural Mission (SACM), for their cooperation and logistic support throughout my M.S. program.

I would like to thank L3-Communications Co. for their belief in the work, funding part of the work, their constant support and enthusiasm. I would like to pay special thanks to David Wiley for his time and valuable feedback since the beginning of this work.

Finally, no one deserves my thanks more than my parents, my mother Hessa and my father Ibrahim. Thanks for the endless support and love throughout my journey in life. I could never have done this without you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Video surveillance is an active research area in computer vision [5]. It received a lot of attention in the last decade [13]. Maritime surveillance is the act of effective detection/recognition of all maritime activities that have impact on economy, security or the environment [18]. Around 80% of all world trade is carried out by sea transport [26]. With the increase in sea transport, there is also increase in illegal acts related to this transport, such as pirate attacks, illegal fishing and immigration, transport of prohibited substances, terrorists attacks specially at ports. Other problem is the collision between sea vehicles specially at the channels and near the coasts and ports.

Estimates show that loss due to piracy might reach US$ 16 billion annually [15]. Attacking civilian and military vehicles is a one way to hurt country's security and economy [17, 38].

Warship Cole had suffered a terrorist attacks, at port Aden in Yemen. 17 people died in this attack [17]. Most of the pirate attacks happen near the coats of Indonesia, Strait of Malacca and Somalia [36].

Manual operation of maritime surveillance systems is not efficient because of stress, fatigue, sleepiness and the limited ability of humans to do certain tasks. Developing automated maritime surveillance systems is essential to avoid dangerous and unwanted events [18, 26, 17].

The use of cameras in maritime surveillance systems has been grown [4]. The cameras are essential to support and assist other systems such as radars and sensors. Cameras are flexible, cheap [38, 24, 21], and can be integrated with almost any platform [13]. Radars have blind zones [38, 24], affected by jamming, clutter and they are expensive [12]. Moreover, radars have low efficiency detecting maritime vehicles made of non-conductive materials [15, 38, 36].

A lot of efforts have been made to develop maritime surveillance systems world wide. Examples include the Autonomous Maritime Surveillance System (AMASS) project which uses FLIR cameras [21]. The Maritime Activity Analysis Workbench (MAAW) [17] and AVITRACK [16] systems are all based on cameras. ARGOS surveillance system is an active system used in Venice to monitor the maritime traffic.

1.2 Overview of the System

In this section the high-level idea about the system will be explained. The idea is that a surveillance cameras will be monitoring a maritime environment which could be an open sea, canal, seaport, pathway, watercourse. These cameras will stream the scene to the observers (shown at Figure 1.1) through a central server. In a traditional process, the observers will have to wathch the maritime environment all the time, to check any unwanted activity. With the system proposed, the unwanted activity is being detected by the servers in the middle of the figure, once this activity is detected the observer will be notified about this suspicious activity. Moreover, the system will classify this activity in terms of the type of the vehicle that entered the scene, by specifying the category type showed in Table 3.1. No tracking is done in this case, since the cameras used are fixed surveillance cameras. Also there will be no bounding box to track this vehicle (even with the fixed camera), since the main goal is to notify the observer about the moving vehicles and to specify their type.

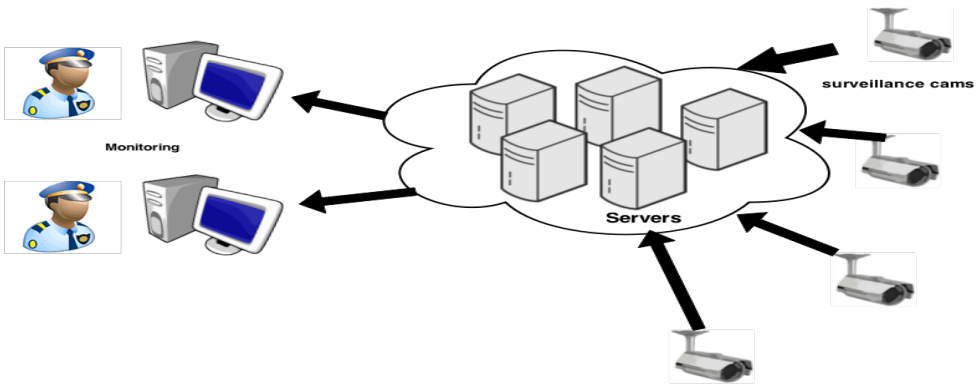

Figure 1.1: System Overview

1.3 Related Work

Maritime environment is a dynamic environment, because of the constant movement of sea waves, movement of clouds, change in lighting due to sun reflection over the sea surface and also due to objects crossing the scene such as birds or airplanes. Because of this dynamic environment, a number of empty scene shots will be classified as scenes that have maritime vehicles in motion (i.e., false positives or FPs) also a number of scenes that have vehicles in motion will be missed and will not be detected (i.e., false negatives or FNs). So, the performance of the initial vehicle detection will affect the overall detection system. A number of detection algorithms are based on frequency [31] and histograms [35]. Some works have focused in Gaussian functions to generate a model for the sea pixels and then detect the vehicle by background subtraction [5, 13, 15, 29]. The constant movement of the ocean waves causes the background subtraction model to fail [15, 38]. The background subtraction model follows the fact that each pixel value follows a normal distribution, or a mixture of normal distributions over periods of time(i.e., night, morning). Then the probability of each pixel to belong to foreground or background will be computed by finding the difference between the pixel value in the tested frame and the pixel mean value of background model, generated by the Gaussian model.

Many works have shown that the efficiency of using a mixture of Gaussians to model the background is less efficient compared to the Gaussian model [26]. Initial detection using optical flow method is not widely used, because it has a high computational complexity [36].

1.4 Contribution

Maritime environment is a dynamic and unpredictable environment. The unpredictability in ocean appearance makes it hard to mathematically model the background. The frames captured by the camera may not have good quality due to weather conditions such as fog, storms or rain. The constant movement of the sea surface, the white foam above the sea surface, the rapid change in lightning due to the sun reflection on the water surface, the movement of clouds and the presence of some moving objects such as airplanes or birds. All of these factors create challenges when modeling such an environment. Because all of these factors, the maritime environment is a challenging environment to predict. One of the main issues, is that because of the above mentioned factors, many False Positives (FP) will be generated (i.e. classifying an empty frame as a frame that has maritime vehicle in motion), which in turn will lower the overall performance of such a system. The system in this work starts by initial motion detector, where it will detect a motion generated by maritime vehicle, and classify each tested frame as a frame that has a maritime vehicle in motion or not. The propose system uses temporal frame differencing method. The problem arises here, is the large number of generated FP frames, mainly caused by the constant movement of sea waves. So, in order to overcome this issue, we smoothed each tested frame using square average filter, so that the effect of the sea surface movement will be less, which in turn will decrease the FP rate. We show that, the performance of the square average filter under the tested scheme (see Figures 4.1 & 4.3), outperforms the performance of the baseline model (i.e. temporal difference with no smoothing filter) and the other tested smoothing

filters such as Gaussian, all tested under the maritime environment. In order to conduct these tests, a new maritime video database was created (see Section 3.1.3).

The second part of this work, is a classification of the frames which were identified to have a motion caused by maritime vehicle (but not other motions, such as the constant movement of sea surface or birds). A subset of ImageNet[1] database was used, categories in this subset have maritime vehicle present in the image (see Section 3.1.1 for examples). Then SVM classifiers were used to classify each vehicle into its category (see Table 3.1 for categories). A total of 13 extracted image features were used (see Table 3.2), all from CloudCV[1][2]. The performance of the SVM classifiers was tested under these 13 image features. Two SVM solving Algorithms were used, Stochastic Gradient Decent (SGD)[32] and Stochastic Dual Coordinate Ascent (SDCA)[33]. At the end of the second part, we show that, the performance of the SDCA under the Decaf features outperforms the performance of SGD in terms of the overall accuracy.

[1]http://www.image-net.org/
[2]http://cloudcv.org/

Chapter 2

Background

In this chapter basic concepts needed to conduct this work will be introduced, starting by explaining the Support Vector Machines (SVMs) in the first section. In the second section the algorithms used to solves the SVMs will be introduced. Finally the last section will show the used filters and image extraction techniques.

2.1 Support Vector Machines

This section's primarily purpose is to provide a brief introduction into SVM. The problems involved in this Research Project are limited to a binary classification of linearly *separable*[1] and *not fully separable*[2] data. The following appendix section is subdivided into two subsections. The first section introduces P_1 whereas the latter deals with P_2.

[1] Denoted as Problem P_1
[2] Denoted as Problem P_2

To simplify the math in section 2.1.1 and section 2.1.2, the basic notation is introduced. Suppose we have a training data set given by $\{\mathbf{x}_i, y_i\}$, where $i = 1, ..., L, y_i \in \{-1, 1\}, \mathbf{x} \in \mathbf{R}^D$. This means that an input vector $\mathbf{x}_i$ consists of D attributes and yields the binary output y_i. We would like to construct a hyperplane $\mathcal{H} = \{\mathbf{x} \in \mathbf{R}^D | \mathbf{w} \cdot \mathbf{x} + b = 0\}$ that separates the two classes[3] given by $y_1 = -1$ and $y_2 = 1$ based on its attributes $\mathbf{x}_i$ in such a way that the *distance* from $\mathcal{H}$ to the closest member of both classes is maximized. The term *Support Vector* is used to describe the set that is closest to the separating hyperplane $\mathcal{H}$. The unknown parameters we would like to determine are therefore $\mathbf{w}$ and b.

The description provided in this chapter is highly based on [14, 7, 8, 6, 3].

2.1.1 Linearly Separable Binary Classification

2.1.1.1 Primal Problem

In Figure 2.1, it is assumed that the given data is perfectly linearly separable. According to Figure 2.1, members of the two classes with the shortest distance to $\mathcal{H}$ will lie on the hyperplanes $\mathcal{H}_1$ and $\mathcal{H}_2$. These hyperplanes are described by $\mathcal{H}_1 = \{\mathbf{x} \in \mathbf{R}^D | \mathbf{w} \cdot \mathbf{x} + b = -1\}$ and $\mathcal{H}_2 = \{\mathbf{x} \in \mathbf{R}^D | \mathbf{w} \cdot \mathbf{x} + b = 1\}$. The distance, in SVM terminology also often referred to as *SVM margin*, between $\mathcal{H}_1/\mathcal{H}_2$ and $\mathcal{H}$ is given by $d_1 = d_2 = 1/\|\mathbf{w}\|$. It is easy to show that *all* points $\{\mathbf{x}_i\}$ are part of two closed, composite *halfspaces* that can be described by the inequality $y_i \cdot (\mathbf{w} \cdot \mathbf{x} + b) - 1 \geq 0, i = 1, ..., L$.

[3] $y_1, y_2 \in y_i$

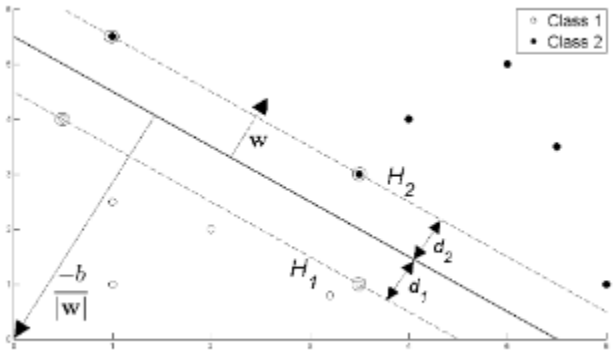

Figure 2.1: Hyperplane through two linearly separable classes [14] Fletcher, Tristan. "Support vector machines explained." Online]. http://sutikno. blog. undip. ac. id/files/2011/11/SVM-Explained. pdf.[Accessed 06 06 2013] (2009). Used under fair use, 2015

Therefore, we can formulate the following optimization problem[4]:

$$\begin{aligned} &\min \|\mathbf{w}\| \\ &\text{subject to} \\ &y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0, \forall i \end{aligned} \tag{2.1}$$

Minimizing $\|\mathbf{w}\|$, on the other hand, is on a par with minimizing $\frac{1}{2}\|\mathbf{w}\|^2$. The optimization problem (2.1) can be restated to a Quadratic Programming (QP) optimization problem:

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{subject to} \\ &y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 \geq 0, \forall i \end{aligned} \tag{2.2}$$

[4]Note that maximizing $1/\|\mathbf{w}\|$ is equivalent to minimizing $\|\mathbf{w}\|$.

With the given QP problem, we obtain the primal problem denoted as L_{P1} where we use the Lagrange multipliers α, where $\alpha_i \geq 0, i = 1, ..., L$.

$$L_{P1} = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{L} \alpha_i y_i (\mathbf{x}_i \cdot \mathbf{w} + b) + \sum_{i=1}^{L} \alpha_i \tag{2.3}$$

2.1.1.2 Dual SVM Problem

As we are interested to find $\mathbf{w}$ and b which minimizes, and α which maximizes Equation (2.3), we consider $\frac{\partial L_{P1}}{\partial \mathbf{w}} = 0, \frac{\partial L_{P1}}{\partial \mathbf{b}} = 0$ given by:

$$\frac{\partial L_{P1}}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i \tag{2.4}$$

$$\frac{\partial L_{P1}}{\partial b} = 0 \Rightarrow \sum_{i=1}^{L} \alpha_i y_i = 0 \tag{2.5}$$

Using Equation (2.4) and (2.5) in the primal problem (Equation (2.3)) generates the dual problem L_{D1} given by:

$$\begin{aligned} &\max \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T \mathbf{H} \alpha \\ &\text{subject to} \\ &\alpha_i \geq 0, \forall i, \qquad \sum_{i=1}^{L} \alpha_i y_i = 0, \end{aligned} \tag{2.6}$$

where $H_{ij} := y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$. Using the fact that each support vector $\mathbf{x}_s$ lies on $\mathcal{H}_1$ or $\mathcal{H}_2$, i.e. fulfills the inequality from (2.2) *with equality*, we can use Equation (2.4) to find an expression

for the bias scalar b. That is

$$b = y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m \cdot \mathbf{x}_s, \tag{2.7}$$

where S[5] corresponds to the set of indices of Support Vectors. Equation (2.7) uses a *single*, *arbitrary* y_s to compute b. In practice, it is preferred to take an average over all N_S Support Vectors in S. Hence, we can compute b according to

$$b = \frac{1}{N_S} \sum_{s \in S} \left(y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m \cdot \mathbf{x}_s \right). \tag{2.8}$$

2.1.2 Not Fully Linearly Separable Binary Classification

2.1.2.1 Primal Problem

An example for data that is not entirely linearly separable is given in Figure 2.2. The halfspaces that were generated by the hyperplanes $\mathcal{H}_1$ and $\mathcal{H}_2$ from section 2.1.1 were modified by a positive slack variable $\xi_i, i = 1, ..., L$. The halfspaces can be described by:

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq +1 - \xi_i \tag{2.9}$$

$$\mathbf{x}_i \cdot \mathbf{w} + b \geq -1 + \xi_i \tag{2.10}$$

$$\xi_i \geq 0, i = 1, ..., L \tag{2.11}$$

[5]Note that S is determined by looking for all indices k where $\alpha_k > 0$.

With the non-negative slack variable, The primal problem can be stated as

$$\begin{aligned} &\min \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{L}\xi_i \\ &\text{subject to} \\ &y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i \geq 0, \qquad \xi_i \geq 0 \qquad \forall i \end{aligned} \tag{2.12}$$

We can formulate a similar Lagrangian primal problem L_{P2} for $P2$ as it was done in Equation (2.3) for $P1$. For the exact derivation, we recommend the interested reader to go through the derivation provided in [14]. The Lagrangian primal problem L_{P2} ($\alpha_i \geq 0, \mu_i \geq 0, i = 1, ..., L$) is given by

$$L_{P2} = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{L}\xi_i + \sum_{i=1}^{L}\mu_i\xi_i - \sum_{i=1}^{L}\alpha_i\left[y_i(\mathbf{x}_i \cdot \mathbf{w} + b) - 1 + \xi_i\right] \tag{2.13}$$

The parameter C can be seen as a trade-off constant between the penalty in terms of the slack variable and the SVM margin.

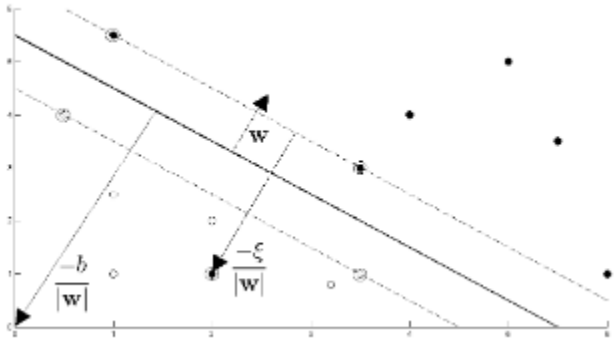

Figure 2.2: Hyperplane through two non-linearly separable classes [14] Fletcher, Tristan. "Support vector machines explained." Online]. http://sutikno. blog. undip. ac. id/files/2011/11/SVM-Explained. pdf.[Accessed 06 06 2013] (2009). Used under fair use, 2015

2.1.2.2 Dual SVM Problem

With the same approach as in section 2.1.1.2, we can derive the dual problem L_{D2} for $P2$ as

$$\max \sum_{i=1}^{L} \alpha_i - \frac{1}{2}\alpha^T \mathbf{H}\alpha$$

$$\text{subject to} \tag{2.14}$$

$$0 \leq \alpha_i \leq C, \forall i, \qquad \sum_{i=1}^{L} \alpha_i y_i = 0,$$

where $H_{ij} := y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$.

The Equation (2.8) that computes b remains unmodified for $P2$ with the exception that the set of Support Vectors S is determined by finding the indices k where $0 < \alpha_k \leq C$.

2.2 Algorithms to Solve SVM

In this sections, the two algorithms used to solve the SVM problem are explained. First Stochastic Gradient Decent (SGD) is explained, which solves the primal problem. In the second part, the Stochastic Dual Coordinate Ascent (SDCA) is explained, which solves the dual SVM problem.

2.2.1 SGD

The Stochastic Gradient Decent (SGD)[32], solve the primal SVM problem shown in equation 2.15, where $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^{m}$ represent the training data set, $\mathbf{x_i} \in \mathbb{R}^n$ and $y_i \in \{+1, -1\}$:

$$\min_{\mathbf{w}} \frac{\lambda}{2} \|\mathbf{w}\|^2 + \frac{1}{m} \sum_{\mathbf{x},y \in S} l(\mathbf{w}; \mathbf{x}, y) \tag{2.15}$$

where

$$l(\mathbf{w}; \mathbf{x}, y) = \max\{0, 1 - y \langle \mathbf{w}, \mathbf{x} \rangle\} \tag{2.16}$$

The SGD pseudo code is shown in Algorithm 1.

2.2.2 SDCA

Stochastic Dual Coordinate Ascent (SDCA)[33], is a linear SVM solver. This method maximizes the dual SVM objective problem shown in Equations 2.17 and 2.18. $\phi_1, ..., \phi_n$ is a

Algorithm 1 The Pegasos Algorithm [32]

Input: S, λ, T
Initialize: Set $\mathbf{w}_1$=0
for t=1,2,...,T **do**
  Choose $i_t \in \{1, ..., |S|\}$ Uniformly at random.
  Set $\eta_t = \frac{1}{\lambda t}$
  if $y_{it} \langle \mathbf{w}_t \mathbf{x}_{it} \rangle < 1$ **then**
    Set $\mathbf{w}_{t+1} \leftarrow (1 - \eta_t \lambda) w_t + \eta_t y_{it} \mathbf{x}_{it}$
  else if $y_{it} \langle \mathbf{w}_t \mathbf{x}_{it} \rangle \geqslant 1$ **then**
    $\mathbf{w}_{t+1} \leftarrow (1 - \eta_t \lambda) \mathbf{w}_t$
  end if
  [**Optional:** $\mathbf{w}_{t+1} \leftarrow min \left\{1, \frac{1/\lambda}{\|\mathbf{w}_{t+1}\|}\right\} \mathbf{w}_{t+1}$]
end for

sequence of convex functions, and $\phi_i^* : \mathbb{R} \rightarrow \mathbb{R}$ is the convex conjugate of ϕ_i:

$$\max_{\alpha \in \mathbb{R}^n} D(\alpha) \tag{2.17}$$

where

$$D(\alpha) = \left[\frac{1}{n} \sum_{i=1}^{n} -\phi_i^*(-\alpha_i) - \frac{\lambda}{2} \left\| \frac{1}{\lambda n} \sum_{i=1}^{n} \alpha_i x_i \right\|^2 \right] \tag{2.18}$$

The pseudocode for SDCA is shown in Algoritm 2.

Algorithm 2 SDCA Algorithm [33]

Let $w^{(0)} = w\left(\alpha^{(0)}\right)$
Iterate: for $t = 1, 2, ..., T$:
  Randomly pick i
  Find $\Delta\alpha_i$ to maximize $-\phi_i^*\left(-\left(\alpha_i^{t-1}\right) + \Delta\alpha_i\right) - \frac{\lambda n}{2}\left\|w^{t-1} + (\lambda n)^{-1}\Delta\alpha_i x_i\right\|$
  $\alpha^t \leftarrow \alpha^{(t-1)}\Delta\alpha_i e_i$
  $w^{(t)} \leftarrow w^{(t-1)} + (n\lambda)^{-1}\Delta\alpha_i x_i$
Output (Averaging option):
  Let $\overline{\alpha} = \frac{1}{T-T_0}\sum_{i=T_0+1}^{T}\alpha^{(t-1)}$
  $\overline{w} = w(\overline{\alpha}) = \frac{1}{T-T_0}\sum_{i=T_0+1}^{T} w^{(t-1)}$
  return $\overline{w}$
Output (Random option):
  Let $\overline{\alpha} = \alpha^{(t)}$ and $\overline{w} = w^{(t)}$ for some random $t \in T_0 + 1, ..., T$
  return $\overline{w}$

2.3 Image Processing

This section focuses on the processing for the images prior to the classification of the frames (i.e. each tested frame will be classified as having maritime vehicle in the scene or not). First subsection will focus on the filters used on the images, second subsection will focus on the image features used prior to the classification step.

2.3.1 Image Filtering

In this subsection a brief overview of each filter used will be shown, starting by Gaussian filter then average, disk and finally motion filter.

2.3.1.1 Gaussian Filtering

Gaussian filter is a two dimensional convolution operator, used usually as a low pass filter to blur images and smooth them. It is a low pass filter that removes the noise or the high frequency details. In a two dimensional space, the filter could be defined as follows:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{\sigma^2}} \tag{2.19}$$

The width and the degree of smoothing of the Gaussian filter can be parametrized be σ. As we increase σ, the Gaussian filter size increases which means higher degree of smoothing for the desired image. Contrariwise, as we decrease the σ value the the size of the Gaussian filter decreases, and hence the number of the neighboring pixels will also decrease, and then the final image will be less smooth compared to the higher σ value.

2.3.1.2 Average Filtering

The average or mean filter is a simple filter. It replaces the pixel value by its neighbors average value. Equation 2.20 define the average filter:

$$h[i, j] = \frac{1}{M} \sum_{k,l \in N} f[k, l] \tag{2.20}$$

Here M represents the total number of the pixels in the square neighborhood N. As we increase the neighborhood size, the pixel value will be averaged over a more neighboring

pixel values, and hence the image will be smoother ans smoother as the size increases.

2.3.1.3 Disk Filtering

The disk filter is a modified version of the original mean or average filter. Instead of having a square filter shape, and then averaging the value of the current pixel over the neighboring pixels values contained in that square, it has a circular or disk shape, and then the value of the current pixel will be the average of the pixels values contained in this circular filter.

2.3.1.4 Motion Filtering

The motion filter also is another modified version of the original average filter, but instead it will average the current pixel value over rows, columns or diagonals, which will indicate the type of motion required. For example, if the average will be taken over the rows then the output image will have horizontal motion (i.e. right to left or left to right). Likewise, if the current pixel value to be averaged over the columns, then the type of motion generated will be vertical (i.e. up to down or down to top). Averaging over the diagonals will generate a mixed motion that has horizontal and vertical axes. The motion filter used in this work is a horizontal motion filter.

2.3.2 Feature Extraction

GIST

GIST descriptors [28] starts by convolving the input image with a 32 Gabor-like filters at 4 different scales and 8 different orientations which will result in a 32 feature map that has the same size as the input image. This feature map will then be divided in 16 regions by a 4x4 grid, and then the feature values will be average across each region. Then finally the 16 regions which were averaged will be concatenated together to form the 512 GIST descriptor.

Histogram of Oriented Gradients (HOG)

Histogram of Oriented Gradients can be used as a feature descriptor for image classification, it was first introduced by [10] on pedestrian images. It is a very popular technique for object detection as well.

The algorithm starts by dividing the input image into sub-images or windows called cells, these cells could be rectangular or circular as shown in figure 2.3. Then the histogram of edge orientation for every pixel within each cell will be accumulated, and finally these histograms will be combined all together. Figure 2.4 shows these steps, Normalizing gamma and colour is an optional step, used usually to make the extracted feature invariant against illumination and shadowing.

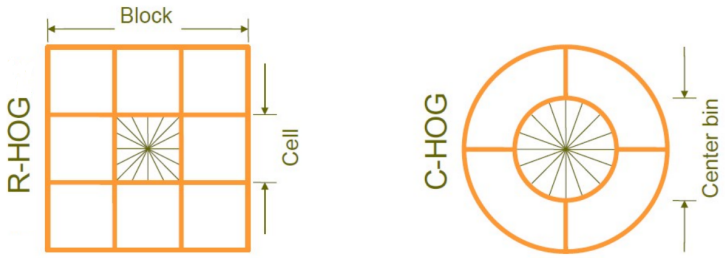

Figure 2.3: Rectangular and Circular cells for HOG [9] Dalal, Navneet. Finding people in images and videos. Diss. Institut National Polytechnique de Grenoble-INPG, 2006. Used under fair use, 2015

The HOG features have many advantages over the feature extraction techniques. HOG is invariant to geometric and photometric transformation[39], moreover the parameters could be tuned to suit different applications, such as human detection [10] and hand gesture recognition [25].

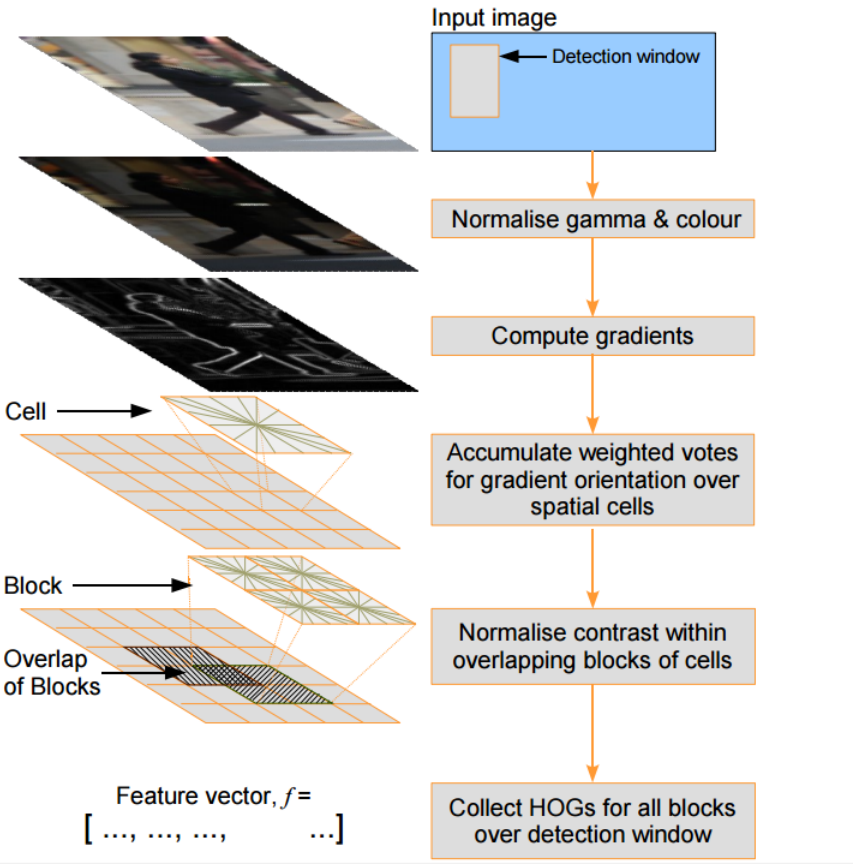

Figure 2.4: HOG steps [9] Dalal, Navneet. Finding people in images and videos. Diss. Institut National Polytechnique de Grenoble-INPG, 2006. Used under fair use, 2015

Decaf

Deep Convolutional Activation Feature (DeCAF) [11], is a deep learning feature extraction technique, that has 7 layers. The layers of this method followed the protocol introduced by [20], plus two more steps added. Firstly to ignore the original image aspect ratio and resize it to 256x256. The second difference, is not adding the data augmentation trick, which will add random multiples of the RGB's principal components of the pixel value over the dataset.

DeCAF has an open source implementation [6], that has been used to train extracted features used in this work. The features extracted are the output of layer 7 of this model.

Remaining Image Features

The remaining used images features include LBP [27], lbphf [2], Line Features [19], SSIM [34], Textton Histograms [30], Geometric Probability Map [22] and SIFT [23]. For more information about the features the reader can see CloudCV[1][7].

2.3.3 Converting from Colormap to Grayscale and Black&White

All the videos tested are colored videos, and in order to follow the main block diagram of this work, these color frames generated by videos will be converted to grayscale based on the weighted sum shown in equation 2.21:

$$GS = 0.2989 * R + 0.5870 * G + 0.1140 * B \tag{2.21}$$

where R,G and B represents the red, green and blue components of the color frame and GS represents the generated grayscale image. The resulting frame from this equation has pixel levels ranging from 0 to 255. Now, in order to convert the resulting grayscale image into a binary value, we need to set a cutoff or threshold that should be in the range from 0 to 255. The threshold is referred to as Thr_{BW}. Each pixel in the grayscale frame that has a value

[6] https://github.com/UCB-ICSI-Vision-Group/decaf-release/
[7] http://cloudcv.org/

less that or equal to Thr_{BW} will be set as a black pixel (i.e. will be assigned a value of 0), on the otherside, each pixel that has a value greater than Thr_{BW} will be considered as a white pixel (i.e. will be assigned a value of 1).

2.3.4 Temporal Difference

The motion in the videos is detect it by finding the difference between two frames, if we define the first binary value frame to be as I_k, and the subsequent frame as I_{k+1}, then the difference can be calculated as shown in equation 2.22

$$I_{d(k,k+1)} = |I_{k+1} - I_k| \tag{2.22}$$

the resulting image $I_{d(k,k+1)}$ is also a binary value image. The white pixels in the resulting image (i.e. pixels that have a value of 1) represent the motion detect it by finding this difference, the total number of pixels that has a value of in each $I_{d(k,k+1)}$ 1 is refereed to as D_{Pix}. Finally the frames are classified to have a motion if D_{Pix} greater than or equal to Thr_{Pix}, otherwise the frame is classified as a no motion frame. Here Thr_{Pix} is a predefined cutoff.

Chapter 3

Data, Software & Evaluation Criteria

This chapter is mainly to describe the tools used in this thesis. The first section describe the used data sets, for both images and videos. Second section lists the software and packages. Finally the last section shows the hardware used to conduct the tests.

3.1 Data

This section is mainly to show the data used in this thesis including the images and videos. First subsection shows a detailed description about the data used. The second subsection describe the videos dataset.

3.1.1 Images

This subsection describes the images used in the classification step in this thesis. The data used for the initial detection for the maritime vehicle will be described in the next subsection. All the images used in this step are originally from ImageNet[1] database and it is available freely for researchers. The classification data set has 1000 categories. The training data for classification has 1.2M images, with an average of 1200 images per category. The validation and test data has 150,000 images. All the images used in this thesis are from the training data, and this training data has been split into two groups, 90% for training and 10% for testing, the categories used are container ship, fireboat, lifeboat, boat paddle, pirate ship, speed boat and submarine. Table 3.1 show the detailed information about the used dataset. Figures 3.1 and 3.2 shows samples of the images.

Table 3.1: Data Table

Category	**Total Size of Images (Megabytes)**	**Total No. of Images**	**No. of Training Images**	**No. of Testing Images**
Container Ship	163	1300	1170	130
Fireboat	153	1300	1170	130
Lifeboat	153	1300	1170	130
Boat paddle	154	1300	1170	130
Pirate ship	125	1118	1007	111
Speedboat	120	1300	1170	130
Submarine	152	1300	1170	130

[1]http://www.image-net.org/

Figure 3.1: Examples from ImageNet[a] (a)Container Ship (b)Fireboat (c)Lifeboat, (d)Paddle boat (e)Pirate Ship

[a]http://www.image-net.org/

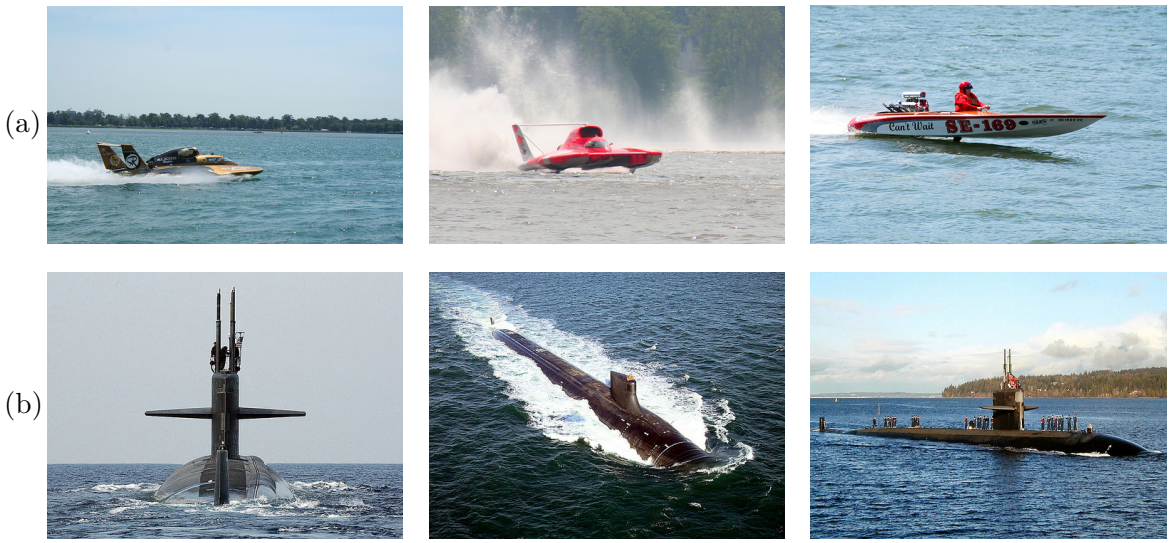

Figure 3.2: More Examples from ImageNet[3](a)Speedboat (b)Submarine

3.1.2 Images Features

The performance of SVM classification step has been tested under 13 features shown in table 3.2 along with with their sizes. The SVM classification performance has been tested in terms of time and accuracy and the detailed testing will be discussed in Chapter 4. All the images features have been collected from CloudCV[1][2]. CloudCV is a cloud service that has many services, one of which is to provide multiple features of the images in ImageNet database. These features are available for free to the researchers and educators.

[3]http://www.image-net.org/

Table 3.2: Features Table

Feature	Total Size (Megabytes)
Decaf	1400
Centre only Decaf	175
Dense SIFT	67
Geometric Probability Map	51
GIST	36
GIST padding	36
HOG2x2	51
LBP padding	36
lbphf	71
Line Features	36
Sparse SIFT histograms	67
SSIM	66
Texton Histograms	141

3.1.3 Videos

All the videos used have been taken in maritime environment with a fixed surveillance camera, so the camera has a fixed scene box and it is not moving in any direction, nor zoom. The scenes of the videos were taken under different maritime environments, some of the videos have an open sea view, and some have a canal and waterway view. The videos were recorded at different times of the day, so some were captured at around sunset, and others were captured during regular day time. All the videos have been collected from Maritime Detection and Tracking data set (MarDT)[4]. After the videos were collected, they have been edited in way such that there are two groups of videos, the first group contains videos with targets moving (i.e. ship or vessels moving in the scene), the second group has an empty scene with no ship or vessels passing in front of the camera, the only movement in this group

[2]http://cloudcv.org/

[4]http://www.dis.uniroma1.it/ labrococo/MAR/

is the movement of the waves of the sea or the canal, and some random birds entering passing in front of the cam.

Table 3.3: Video Dataset

Total Videos Size (Megabytes)	269
Total No. of Videos	85
No. of Videos with Maritime Vehicle in Motion	53
No. of Videos with no Motion	32
Frame Rate (frame/sec.)	9.99
No. of Training Videos	61
No. of Testing Videos	24
Resolution	1024x576
Videos Type	.ogv
Video Format	RGB24

3.2 Software

All of this work results have been conducted under MATLAB environment, and this section will introduce other used packages. VLFeat [37][5] is an open source implementation written C and it has a MATLAB wrapper. It has many Machine Learning implementation such as Fisher Vector, SIFT, k-means, quick shift superpixels and large scale SVM. VLFeat is used in this work to train the SVM classifiers in the second step, the SGD & SDCA are part of the VLFeat implementation to solve the large scale SVM. Also DeCAF[20][6] implementation was used to extract the DeCAF features used in the seconed step of this work, the library is written in C++ and has python and MATLAB wrappers.

[5] http://www.vlfeat.org/

[6] https://github.com/UCB-ICSI-Vision-Group/decaf-release/

3.3 Evaluation Criteria

The evaluation for videos in Table 3.3 starts by dividing the videos into two groups. The first group has videos with maritime vehicles in motion. The second group has videos with motion generated by the sea waves or movement of the clouds or caused by birds crossing the scene or the reflection of the sun over the water surface. Each tested video has a certain number of tested frames based on the sampling rate used. Then for each video the number of correctly and incorrectly classified frames are calculated. The correctly classified frames for a video that has a motion are referred as True Positive(TP), and the incorrectly classified frames for the same group are called False Negative(FN). A widely used formula to measure the performance of a classification algorithm is the Recall and Precision. Recall measures the performance for the positive classifier (i.e. videos that have motion). The Precision is used to find the percentage of the TP over the total frames that was classified as positive. The Recall is also known as True Positive Rate and the precision as Positive Predictive Value (PPV). Precision and recall can be found using Equations 3.1 & 3.2

$$Recall = TPR = \frac{No.\,of\ Positive\,Frames\,Classified\,Correctly}{Total\,No.\,of\ Positive\,Frames} = \frac{TP}{TP + FN} \tag{3.1}$$

$$Precision = PPV = \frac{No.\,of\,Positive\,Frames\,Classified\,Correctly}{Total\,No.\,of\,Frames\,Classified\,as\,Positive} = \frac{TP}{TP+FP} \quad (3.2)$$

Another Known measure for the performance of classifier are the False Positive Rate (FPR) and the False Negative Rate (FNR), they can be found using Equations 3.3 & 3.4

$$FPR = \frac{No.\,of\,Negative\,Frames\,Classified\,as\,Positive}{Total\,No.\,of\,Negative\,Frames} = \frac{FP}{TN+FP} \quad (3.3)$$

$$FNR = \frac{No.\,of\,Positive\,Frames\,Classified\,as\,Negative}{Total\,No.\,of\,Positive\,Frames} = \frac{FN}{TP+FN} \quad (3.4)$$

The Confusion Matrix shown in Table 3.4 clarifies the idea where the actual class shown in columns and the predicted class by the classifier shown in rows.

Table 3.4: Confusion Matrix

		Actual Class	
		Motion	No Motion
Predicted Class	Motion	TP	FP
	No Motion	FN	TN

Chapter 4

Results & Analysis

4.1 Overview

In this Chapter all the results of the tests will be shown. An overview of the surveillance system design will be shown in the first section. In the second section the results and analysis of the classifier step will be shown. Finally, in the last section the results for initial motion detection will be presented.

4.2 System Design

The system has two main steps. The first step main goal is to detect a motion in the scene. Since the surveillance camera is likely to have the same scene at all times(i.e. because the camera is fixed), except when a maritime vehicle enters the scene. The initial detector

is the first step that has a main goal to optimize the performance in terms of time (i.e. processing time), because taking each frame and classify it using the classifier will overload the system and will make impractical. The second step is mainly a classifier that takes the frames detected to have motion and passed by the first step. Only frames which have been identified to have a motion by the first step will be used in the classifier step to identify the type of object in the scene (i.e. fireboat, speedboat, lifeboat, etc.). The reader is advised to read Chapter 3 for a detailed information about the dataset used. Figure 4.1 shows an overview of the steps.

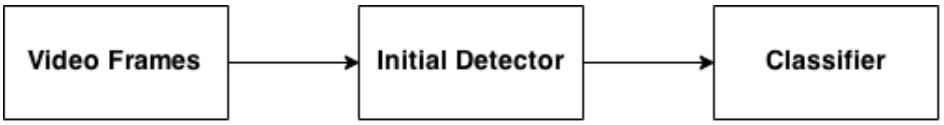

Figure 4.1: Overview of the steps

4.3 Classifier Results

In this section, the analysis and performance of image classification will be shown and evaluated. All the classification for the images is done using SVM (described in chapter 2). SGD and SDCA were both used as training algorithms for SVM classifiers. The SVM classifiers have been trained for 13 image features described in Chapter 2. The C value for all the SVM classifiers is 100. Also each category set (i.e. container ship, boat, etc.) has it's own classifier. So for each category set 1 vs. all SVM classifier has been trained. For example, when training the SVM classifier for fireboat, the only positive set will be the fireboat, and

all the other sets (i.e. all categories such as lifeboat, speedboat and submarine, but not the fireboat) will be considered as negative examples. For for each image feature we will have 7 SVM classifiers. And we have a total of 13 extracted features. So, a total of 7x13 SVM classifiers have been trained. SVM classification step will be used only if a maritime vehicle motion is detected by the initial motion detector (see Figure 4.1). So, if any frame found to have motion it will go the SVM classifier step to identify the category of the object detected. This step starts by taking the frame that have been identified to have motion, then it will extract the features of that frame, and then will be tested under all the 7 SVM classifiers for each image feature set. Each classifier will assign a score to the input frame. After all the scores have assigned, the object will be classified under the category that has the maximum score. The steps are shown in Figure 4.2.

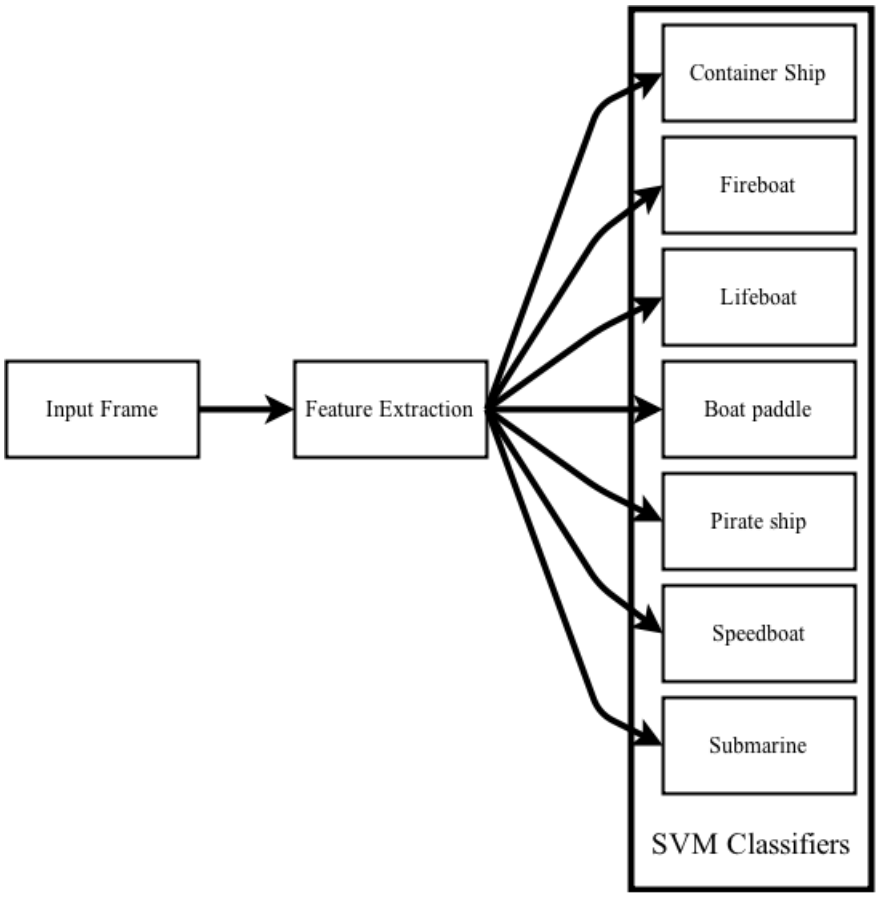

Figure 4.2: Classification Steps

Table 4.1 shows a comparison between all the features tested under the two algorithms used (i.e. SGD and SDCA). The features that have the lowest error are Decaf and Decaf with the Centre only option. However Decaf features (not the decaf with centre option) do not perform well in terms of training time compared to the other features. All the tests have been trained 5 times for each SVM classifier, and then averaged over these 5 trains. The error rate changes as we train the SVM classifier again, this is due to the fact that both the SGD and SDCA are all stochastic algorithms, and their error rate changes from one training round to another. Likewise, the training time for the two algorithms was averaged over the

5 training rounds.

The error rate, is calculated by dividing the number of test images that were classified incorrectly over the total number of the tested images for each image feature set(i.e. Decaf, SIFT, etc.). The training set represents 90% of the total set, and the remaining 10% of the images were used for testing. More detailed information about the dataset could be found in Table 3.1 and Table 3.2. Only the top score is considered as the object category. So for example if we have input image of fireboat and the SVM top score was assigned to speedboat and the second top score was assigned to fireboat, then in this case the image will be classified as speedboat and will be counted in the error rate.

Table 4.1: Classifier Results

Feature	**SGD Training Time**	**SDCA Training Time**	**SGD Error**	**SDCA Error**
Decaf	29.592	42.916	0.11108	0.08238
Centre only Decaf	3.253	4.884	0.10552	0.08036
Dense SIFT	3.6675	4.5638	0.57238	0.81438
Geometric Probability Map	0.2113	0.2648	0.71562	0.72414
GIST	0.3994	0.5354	0.62604	0.61862
GIST padding	0.3829	0.5153	0.62626	0.64736
HOG2x2	3.6996	4.8775	0.53174	0.77732
LBP padding	0.6689	0.8578	0.72302	0.7248
lbphf	0.4397	0.6164	0.65412	0.66128
Line Features	0.2021	0.2484	0.76724	0.7917
Sparse SIFT histograms	1.4602	1.9873	0.56228	0.53626
SSIM	3.5227	4.6169	0.58452	0.81662
Texton Histograms	5.7285	8.1944	0.70146	0.78452

4.4 Initial Detection Results

The goal of the initial motion detection step, is to find the initial motion generated by a moving maritime vehicle in the scene, in this case the system computational complexity will be less compared to scheme where each sampled frame will be classified directly (i.e. going directly to the second step in Figure 4.1). It is faster because in this step we avoid the image feature extraction, which is a time consuming step. The initial motion that needs to be identified is the motion of a maritime vehicle that enters the scene of the fixed surveillance camera. Once a frame is identified to have motion it will go to the classifier step as shown in Figure 4.1. The videos used in this section were split in to two categories, the first category has videos that contain a moving maritime vehicle, these moving vehicles are identified in Table 3.1, other types of movements are not included in this category (i.e. movement of birds entering the scene or the clouds). The second category contains videos that have no maritime vehicles in motion, the only types motion in this case are the motions of the sea, canal or waterway waves, and the motion of birds crossing in the scene. The details about the videos could be found in Table 3.3. The initial motion is detected by sampling two consecutive frames. For the time gap, between these two consecutive frames, or the sampling rate (i.e. the time at which a frame will be taken as a sample) multiple sample rates 2,4,6,8,10 were tested, for example 2 means a frame sample/(2 sec.), the same applies for the rest sampling rates. The two frames will then be filtered using different filters (rectangular average filter, circular average filter, Gaussian filter, motion filter), the reader can find more details about these filters in Chapter 2. Now the absolute difference between these two

filtered frames will be calculated. After that, this absolute difference (colored image) will be converted into a grayscale image based on equation 2.21. Finally the resultant grayscale image will be converted into Black&White image, based on the fact that, any pixel that has a value greater than Thr_{BW} will be white, and any pixel value less than Thr_{BW} will be considered as black. Multiple thresholds Thr_{BW} were tested (13,26,52,77). The number of pixels that differ in two consecutive frame are the pixels that have a value of 1 (i.e. white), the frame is considered to have a moving maritime vehicle if the number of pixels that differ (i.e. have value of 1) exceeds a predefined threshold called Thr_{Pix}. A diagram showing these steps is illustrated in Figure 4.3 .The threshold Thr_{Pix} was tested under different values (0,1,10,20,30,40,50,60,70,80,90,100,200,300,400,500,600,700,800,900,1000,10000,100000). Tables 4.2 & 4.3 shows the difference between two consecutive frames under different frame rates, and then finding the mean and the standard deviation for these differences (i.e. the white pixels generated in last step, see Figure 4.3) for both categories (i.e. with motion and without motion).

Table 4.2: Mean and Variance of the Difference Between Two consecutive Frames for Videos without Moving Object

Frame Sample Rate	**Mean**	**Standard Deviation**
1	29.44	73.77
2	50.03	117.95
3	53.82	123.05
4	48.01	105.54
5	59.67	139.44
6	58.45	145.38
7	58.73	126.36
8	62.79	142.90
9	54.45	112.12
10	57.21	141.01

Table 4.3: Mean and Variance of the Difference Between Two consecutive Frames for Videos with Moving Object

Frame Sample Rate	**Mean**	**Standard Deviation**
1	560.1	1170.9
2	806.3	1600.9
3	881.2	1796.0
4	907.4	1760.9
5	975.6	1900.4
6	1023.4	1957.7
7	1050.6	2151.2
8	1066.5	1967.9
9	1052.6	2096.5
10	1084.7	2168.9

Now, the goal is to find the filter that performs the best under the scheme illustrated in Figure 4.3 in terms of time performance and accuracy. Figures 4.5, 4.6, 4.7 and 4.8 shows the ROC curve behaviour for different filters while each figure tested under fixed Thr_{BW} and multiple Thr_{Pix}. Figure 4.4 shows the optical flow ROC curve behaviour under multiple thresholds.

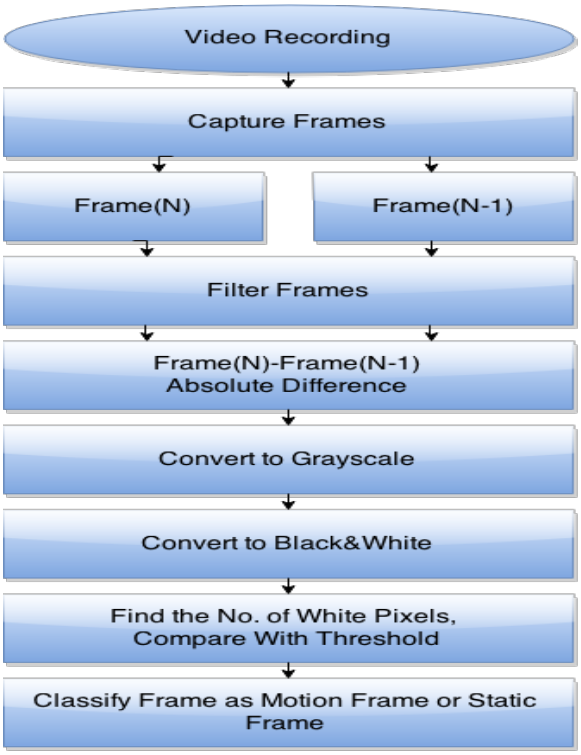

Figure 4.3: Motion Detection Steps

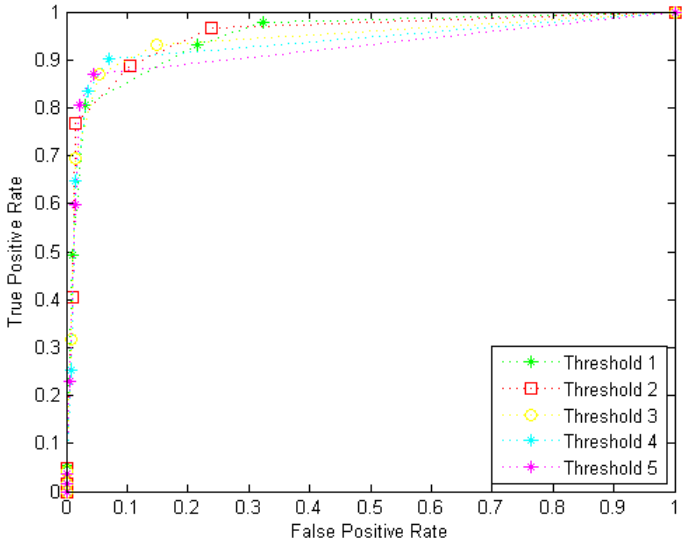

Figure 4.4: Optical flow ROC curve behaviour under multiple thresholds

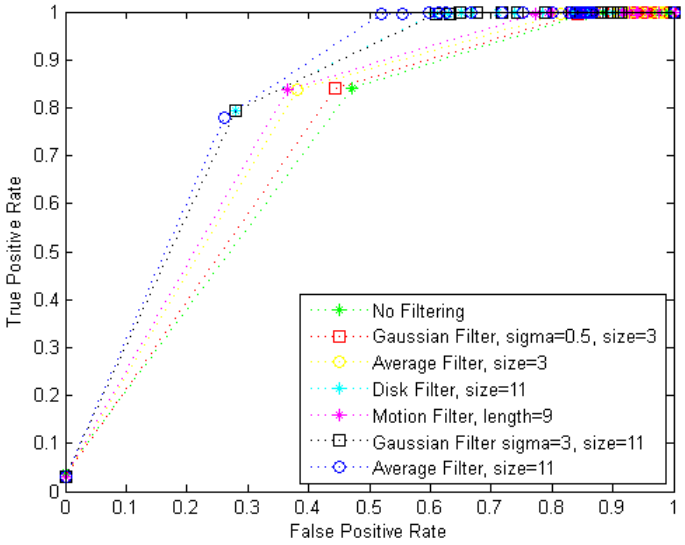

Figure 4.5: ROC curve with $Thr_{Pix} = 13$

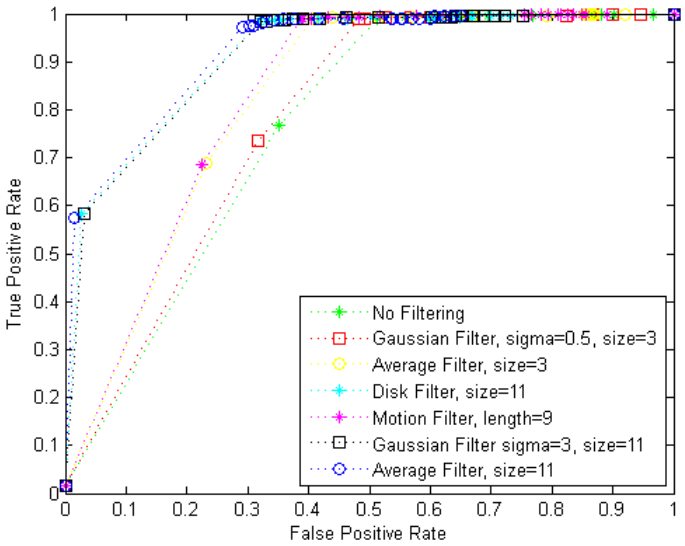

Figure 4.6: ROC curve with $Thr_{Pix} = 26$

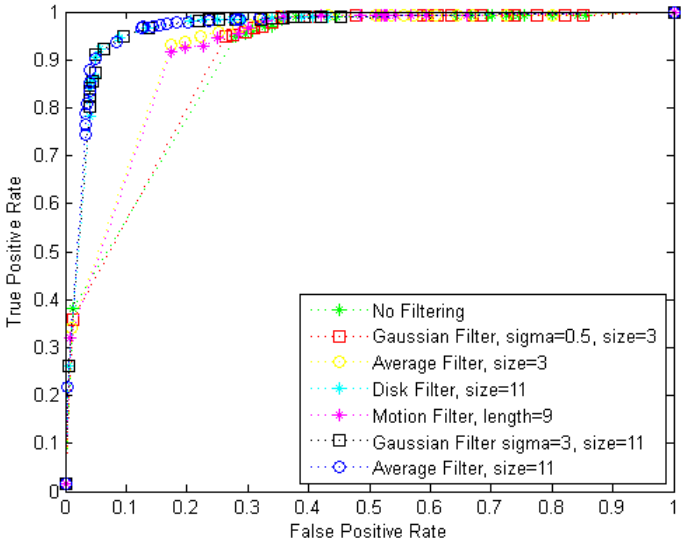

Figure 4.7: ROC curve with $Thr_{Pix} = 52$

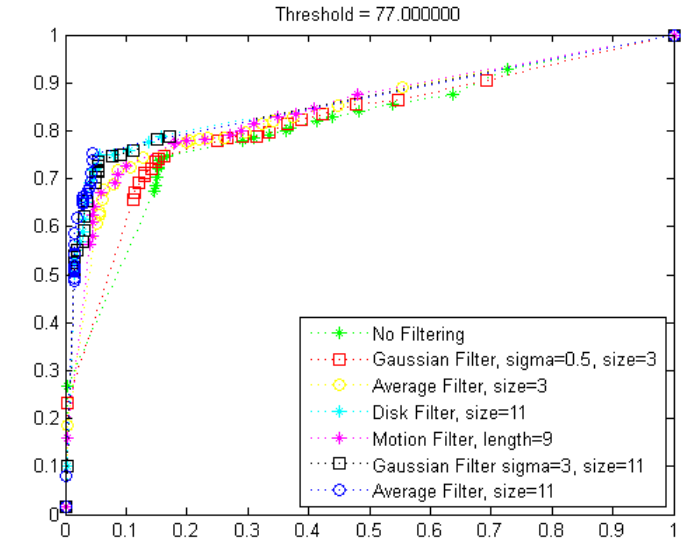

Figure 4.8: ROC curve with $Thr_{Pix} = 77$

Figure 4.9 shows the optical flow precision & recall curve behaviour under multiple thresholds. Figures 4.10, 4.11, 4.12 and 4.13 shows the precision and recall curves for different

filters while each figure tested under a fixed Thr_{BW} and multiple Thr_{Pix}. The time performance of the different filters can be found in Figure 4.14. In order to have more robust results, each test case (i.e. fixed Thr_{BW} and fixed Thr_{Pix}) was tested under 5 sampling rates (2,4,6,8,10) then all of the time and accuracy results were averaged over theses 5 sampling rates.

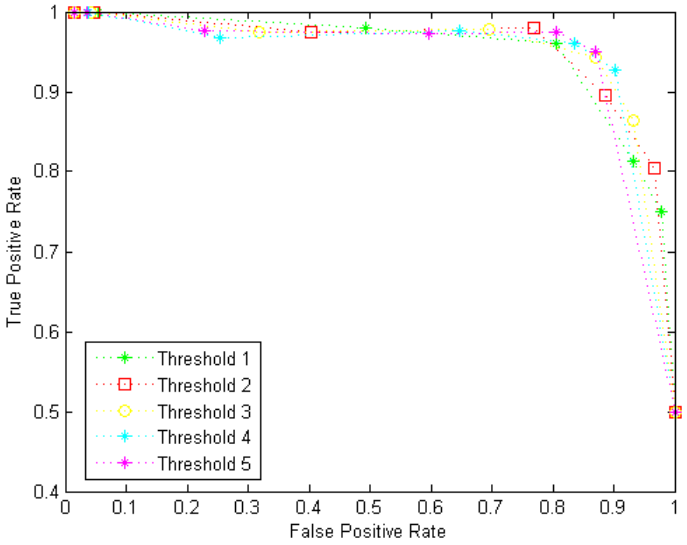

Figure 4.9: Optical flow Precision & Recall curve behaviour under multiple thresholds

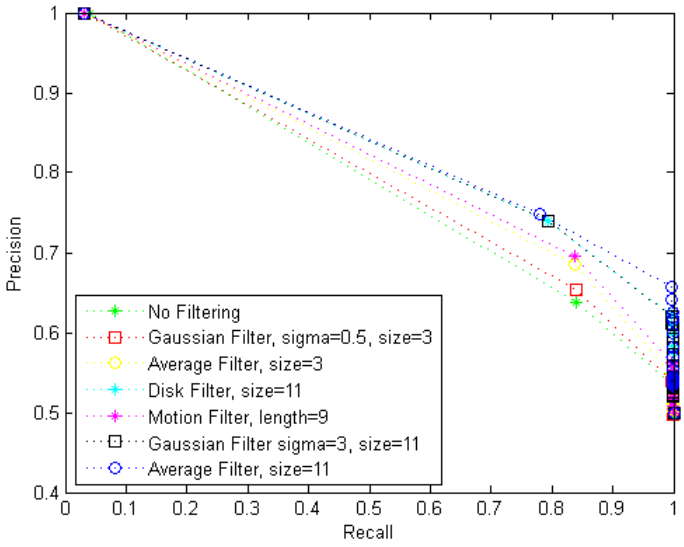

Figure 4.10: Precision Recall curve with $Thr_{Pix} = 13$

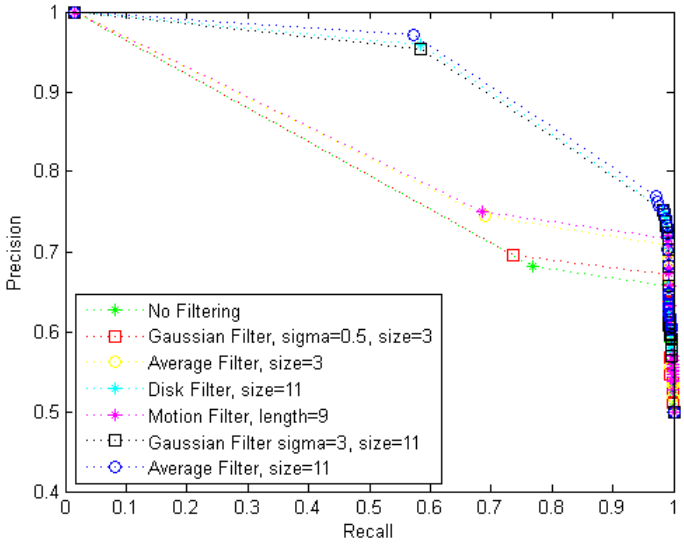

Figure 4.11: Precision Recall curve with $Thr_{Pix} = 26$

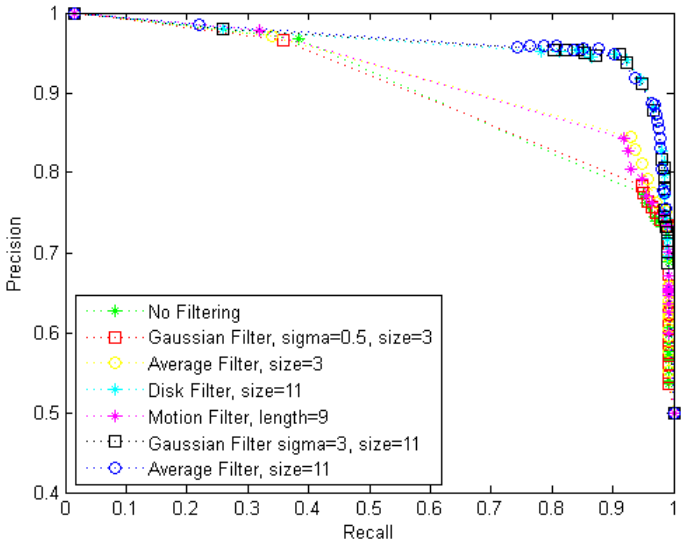

Figure 4.12: Precision Recall curve with $Thr_{Pix} = 52$

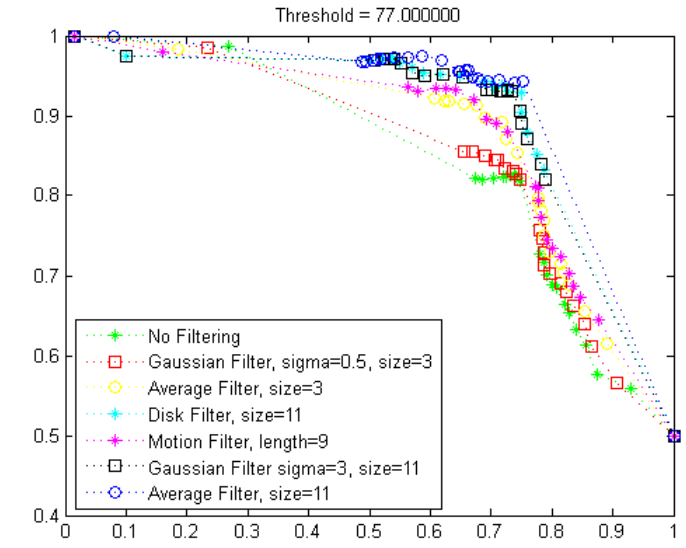

Figure 4.13: Precision Recall curve with $Thr_{Pix} = 77$

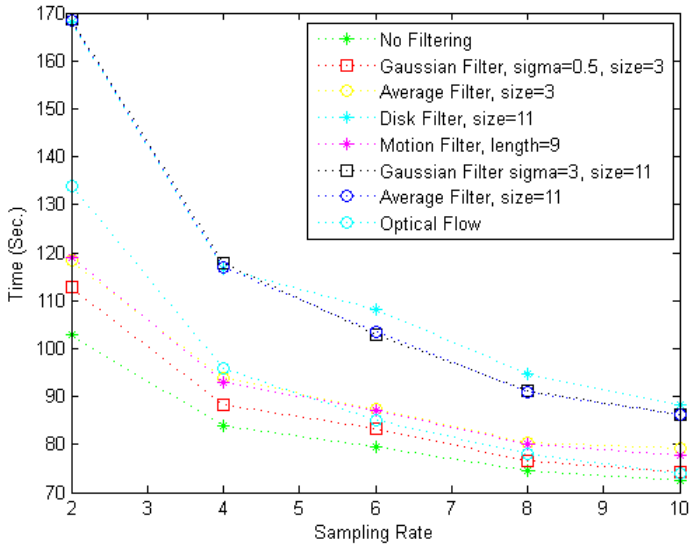

Figure 4.14: Time comparison between different methods

Chapter 5

Conclusion

The thesis is divided mainly into two parts, in the first part, the maritime vehicle classification part we classify the images using SVM classifiers. The results have been tested under 13 image features, of which, the Decaf features outperforms the other features in terms of accuracy, even though it does not perform well in terms of the training time. However, the training time is not a big factor in such a system, since it will be done only once, the Decaf and the Decaf with Centre only option have almost the same results in terms of accuracy. Moreover, the SDCA solving algorithm performs notably better than the SGD algorithm in terms of the accuracy, when combined with the Decaf features, achieving around 2 3% less error compared to SGD. However, the SGD algorithm performs better when tested under the other images features other than the Decaf.

As for the second part of the thesis, or the motion detection part, the scheme for the different filters is the same, as shown in figure 4.3, the only difference is the filter type, also we

compare the results with the optical flow, widely used method for motion detection. Finding the absolute difference between frames without the use of filter is the fastest compared to the other schemes, even when compared to the optical flow, as one can see in Figure 4.14. The temporal frame difference using the averaging filter achieved the best results in terms of the overall accuracy, this could be seen in the ROC curves in Figures 4.4,4.5,4.6,4.7 & 4.8, and the precision and recall curves in Figures 4.9,4.10,4.11,4.12 & 4.13. The average filter and the Gaussian filter have a similar time performance as shown in Figure 4.14.

Appendix A

Appendix: ROC, Precision & Recall Curves

A.1 ROC Curves

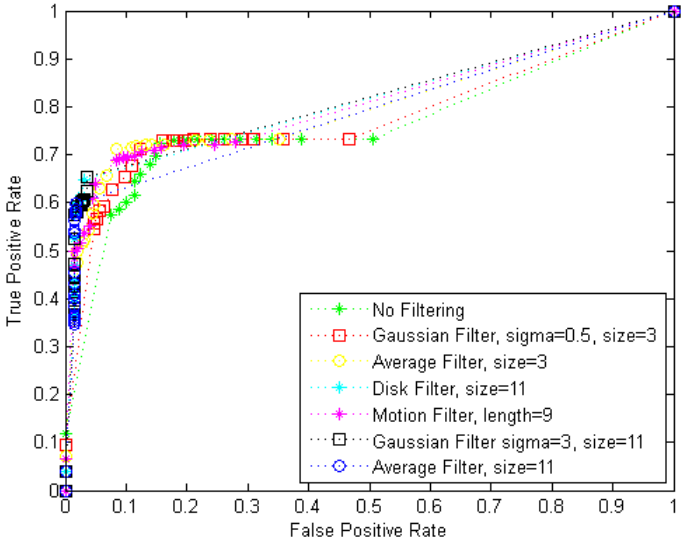

Figure A.1: ROC curve with $Thr_{Pix} = 103$

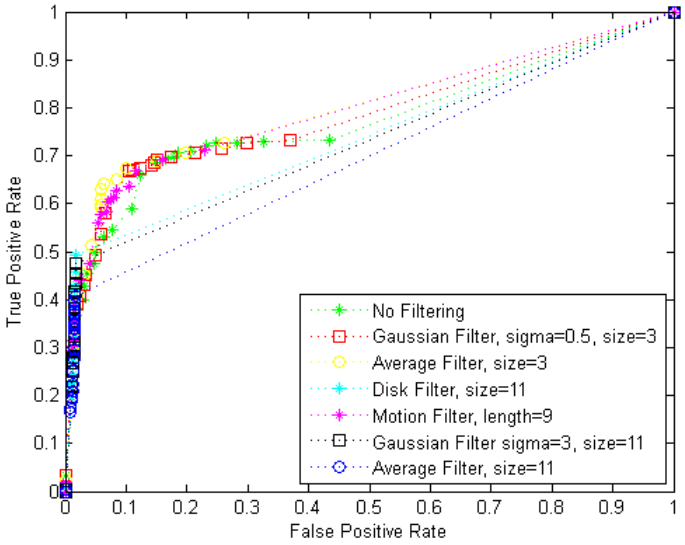

Figure A.2: ROC curve with $Thr_{Pix} = 128$

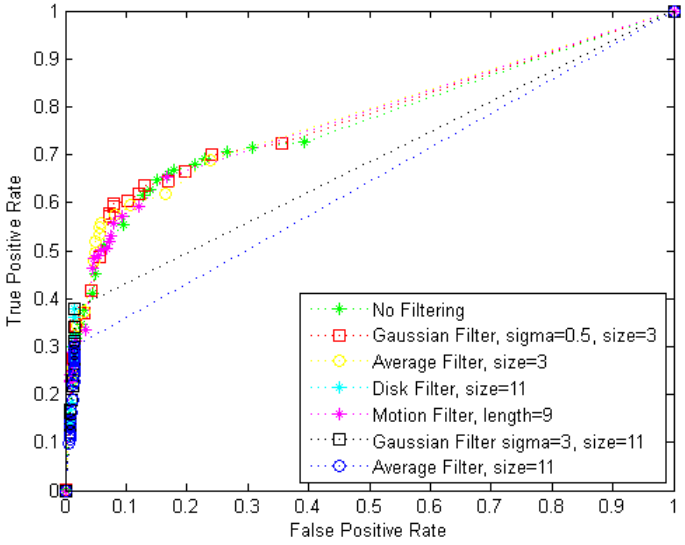

Figure A.3: ROC curve with $Thr_{Pix} = 140$

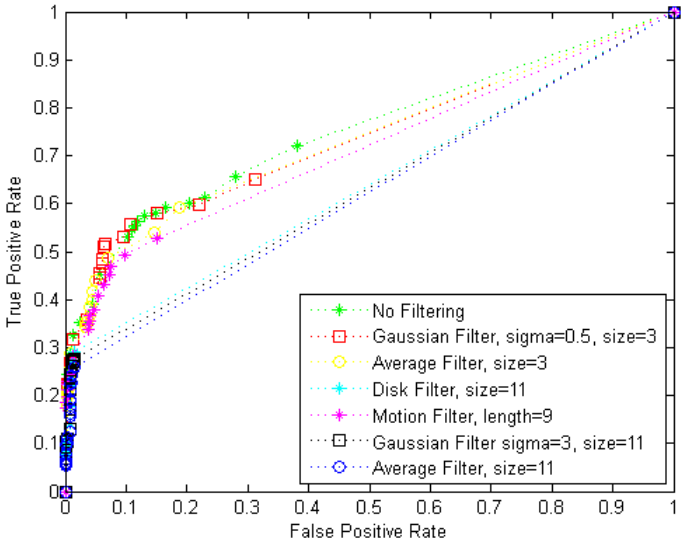

Figure A.4: ROC curve with $Thr_{Pix} = 154$

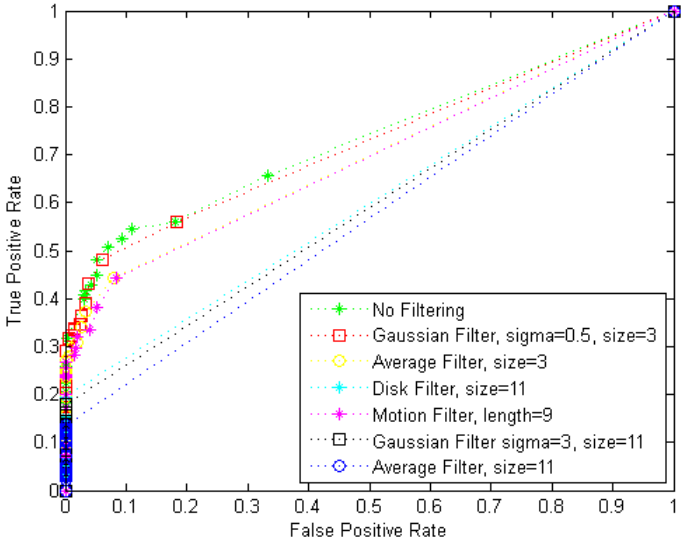

Figure A.5: ROC curve with $Thr_{Pix} = 180$

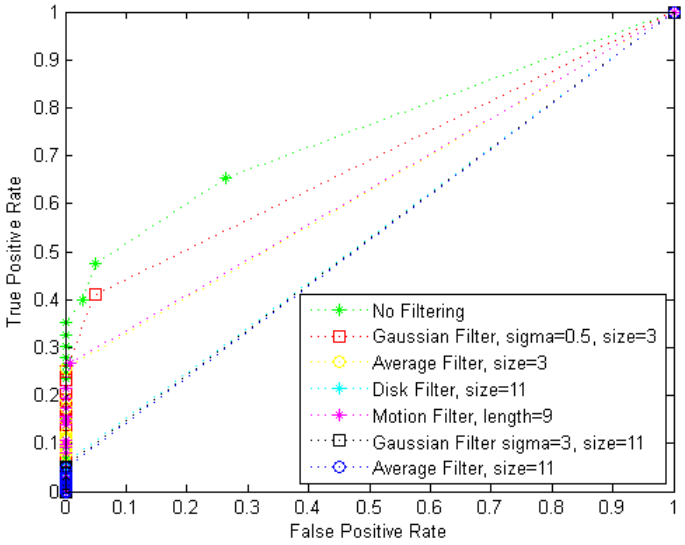

Figure A.6: ROC curve with $Thr_{Pix} = 205$

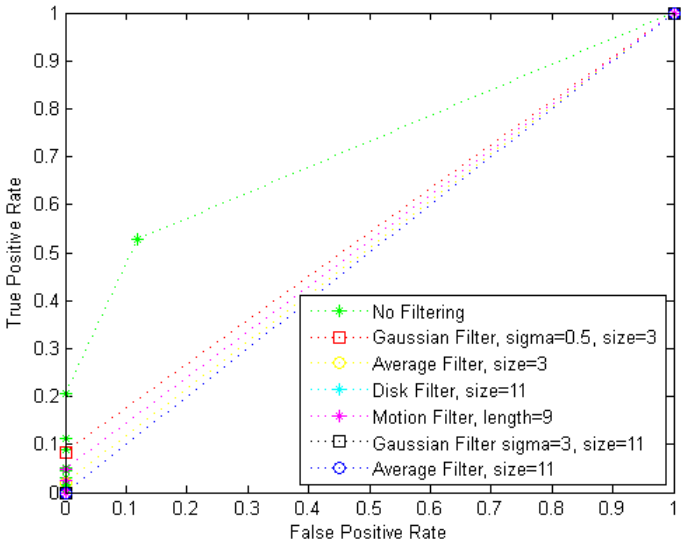

Figure A.7: ROC curve with $Thr_{Pix} = 231$

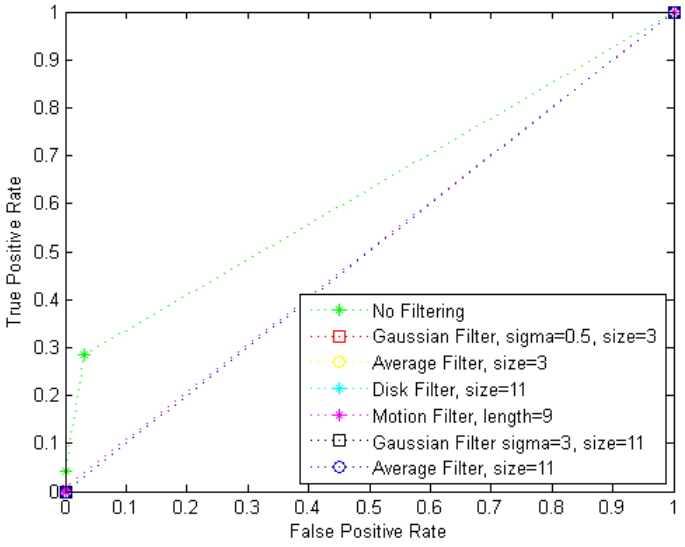

Figure A.8: ROC curve with $Thr_{Pix} = 244$

A.2 Precision Recall Curves

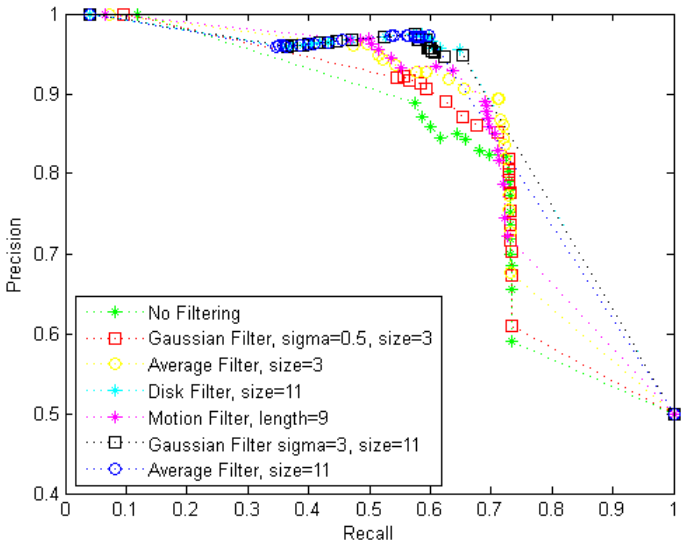

Figure A.9: Precision Recall curve with $Thr_{Pix} = 103$

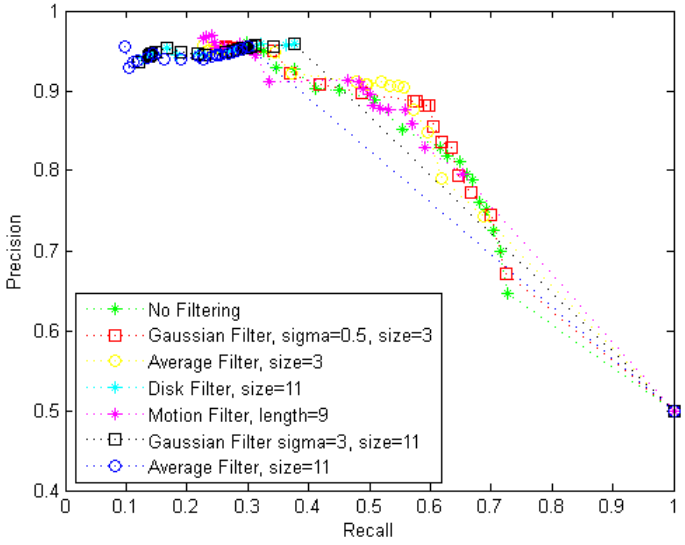

Figure A.10: Precision Recall curve with $Thr_{Pix} = 140$

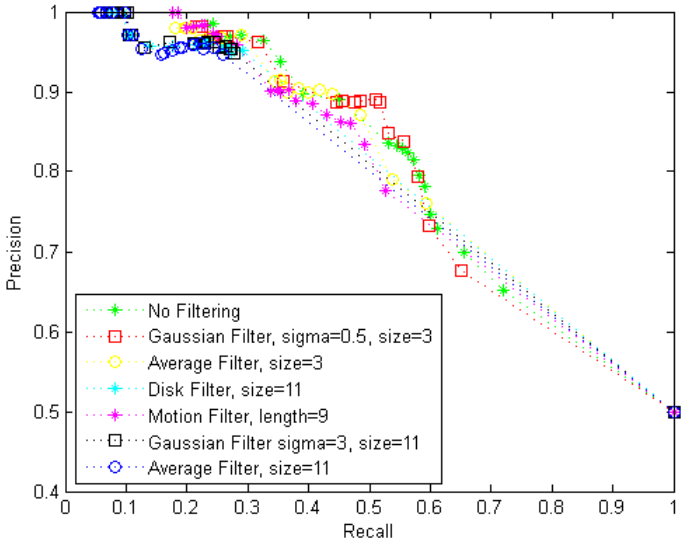

Figure A.11: Precision Recall curve with $Thr_{Pix} = 154$

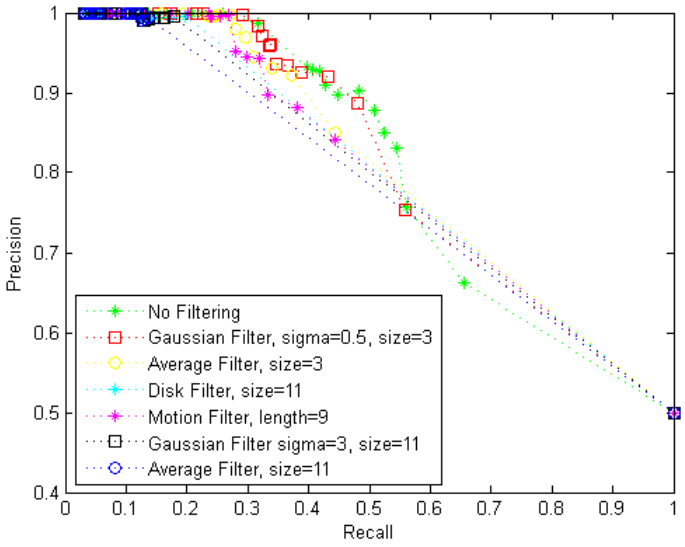

Figure A.12: Precision Recall curve with $Thr_{Pix} = 180$

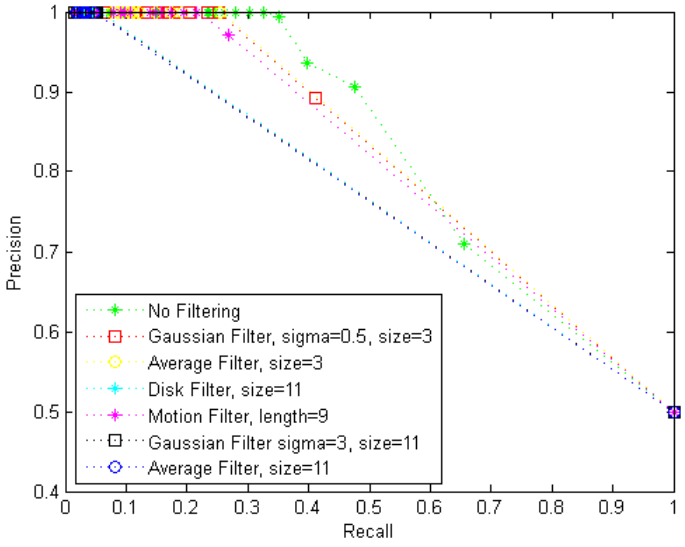

Figure A.13: Precision Recall curve with $Thr_{Pix} = 205$

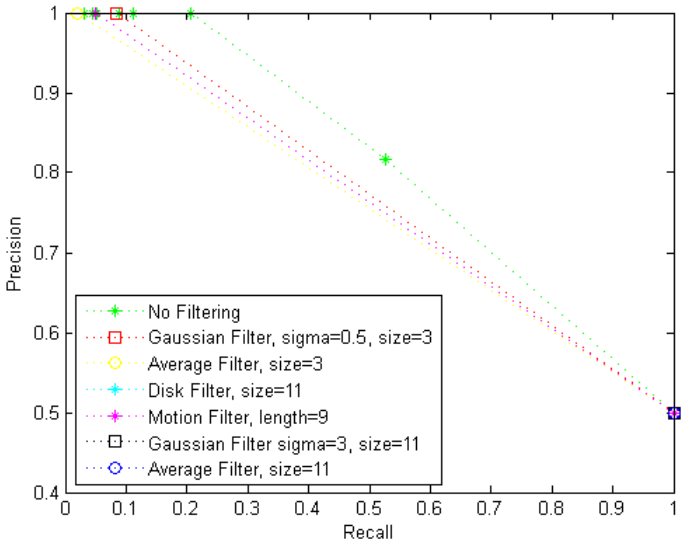

Figure A.14: Precision Recall curve with $Thr_{Pix} = 231$

Bibliography

[1] H. Agrawal, N. Chavali, Mathialagan C., Y. Goyal, A. Alfadda, , P Banik., and D. Batra. Cloudcv: Large-scale distributed computer vision as a cloud service, 2013.

[2] Timo Ahonen, Jiří Matas, Chu He, and Matti Pietikäinen. Rotation invariant image description with local binary pattern histogram fourier features. In *Image Analysis*, pages 61–70. Springer, 2009.

[3] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[4] Domenico Bloisi, L Iocchi, M Fiorini, and G Graziano. Automatic maritime surveillance with visual target detection. In *Proc. of the International Defense and Homeland Security Simulation Workshop (DHSS)*, pages 141–145, 2011.

[5] Domenico Bloisi and Luca Iocchi. ArgosâĂŤa video surveillance system for boat traffic monitoring in venice. *International Journal of Pattern Recognition and Artificial*

Intelligence, 23(07):1477–1502, 2009.

[6] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2:121–167, 1998.

[7] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[8] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[9] Navneet Dalal. *Finding people in images and videos*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2006.

[10] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[11] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.

[12] Sergiy Fefilatyev. *Detection of marine vehicles in images and video of open sea*. PhD thesis, University of South Florida, 2008.

[13] Sergiy Fefilatyev, Dmitry Goldgof, Matthew Shreve, and Chad Lembke. Detection and tracking of ships in open sea with rapidly moving buoy-mounted camera system. *Ocean Engineering*, 54:1–12, 2012.

[14] Tristan Fletcher. Support vector machines explained. *Online]. http://sutikno. blog. undip. ac. id/files/2011/11/SVM-Explained. pdf.[Accessed 06 06 2013]*, 2009.

[15] Duncan Frost and Jules-Raymond Tapamo. Detection and tracking of moving objects in a maritime environment using level set with shape priors. *EURASIP Journal on Image and Video Processing*, 2013(1):1–16, 2013.

[16] Florent Fusier, Valéry Valentin, François Brémond, Monique Thonnat, Mark Borg, David Thirde, and James Ferryman. Video understanding for complex activity recognition. *Machine Vision and Applications*, 18(3-4):167–188, 2007.

[17] Kalyan Moy Gupta, David W Aha, Ralph Hartley, and Philip G Moore. Adaptive maritime video surveillance. In *SPIE Defense, Security, and Sensing*, pages 734609–734609. International Society for Optics and Photonics, 2009.

[18] Samira Kazemi, Shahrooz Abghari, Niklas Lavesson, Henric Johnson, and Peter Ryman. Open data for anomaly detection in maritime surveillance. *Expert Systems with Applications*, 40(14):5719–5729, 2013.

[19] Jana Košecká and Wei Zhang. Video compass. In *Computer VisionâĂŤECCV 2002*, pages 476–490. Springer, 2002.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[21] W Kruger and Zigmund Orlov. Robust layer-based boat detection and multi-target-tracking in maritime environments. In *Waterside Security Conference (WSS), 2010 International*, pages 1–7. IEEE, 2010.

[22] Jean-François Lalonde, Derek Hoiem, Alexei A Efros, Carsten Rother, John Winn, and Antonio Criminisi. Photo clip art. *ACM transactions on graphics (TOG)*, 26(3):3, 2007.

[23] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[24] Haiying Liu, Omar Javed, Geoff Taylor, Xiaochun Cao, and Niels Haering. Omni-directional surveillance for unmanned water vehicles. In *The Eighth International Workshop on Visual Surveillance-VS2008*, 2008.

[25] David Fernández Llorca, Ignacio Parra, Miguel Ángel Sotelo, and Gerard Lacey. A vision-based system for automatic hand washing quality assessment. *Machine Vision and Applications*, 22(2):219–234, 2011.

[26] Rodrigo Da Silva Moreira, Nelson Francisco Favilla Ebecken, Alexandre Soares Alves, Frédéric Livernet, and Aline Campillo-Navetti. A survey on video detection and tracking of maritime vessels.

[27] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[28] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[29] Nuno Pires, Jonathan Guinet, and Elodie Dusch. Asv: an innovative automatic system for maritime surveillance. *Navigation*, 58(232):1–20, 2010.

[30] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17. IEEE, 2003.

[31] Jason G Sanderson, Martin Kenneth Teal, and Tim J Ellis. Characterisation of a complex maritime scene using fourier space analysis to identify small craft. In *Image Processing and Its Applications, 1999. Seventh International Conference on (Conf. Publ. No. 465)*, volume 2, pages 803–807. IET, 1999.

[32] Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical programming*, 127(1):3–30, 2011.

[33] Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *The Journal of Machine Learning Research*, 14(1):567–599, 2013.

[34] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[35] AA Smith and MK Teal. Identification and tracking of maritime objects in near-infrared image sequences for collision avoidance. 1999.

[36] Zygmunt L Szpak and Jules R Tapamo. Maritime surveillance: Tracking ships inside a dynamic background using a fast level-set. *Expert systems with applications*, 38(6):6669–6680, 2011.

[37] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008.

[38] Hai Wei, Hieu Nguyen, Prakash Ramu, Chaitanya Raju, Xiaoqing Liu, and Jacob Yadegar. Automated intelligent video surveillance system for ships. In *Proceedings of SPIE*, volume 7306, page 73061N, 2009.

[39] Yixiao Yun. Analysis and classification of object poses-using visual/infrared images and feature fusion. 2011.