ISCTE IUL

Instituto Universitário de Lisboa

Detection and identification of registration and fishing gear in vessels

Jorge Miguel de Albuquerque Branquinho

Dissertation submitted as a partial requirement to obtain a Master's degree in:

Information Systems and Knowledge Management

Professor:

PhD João Carlos Amaro Ferreira

ISCTE-IUL

September, 2017

Contents

Abstract*—Illegal, unreported and unregulated (IUU) fishing is a global menace to both marine ecosystems and sustainable fisheries. IUU products often come from fisheries lacking conservation and management measures, which allows the violation of bycatch limits or unreported catching. To counteract such issue, some countries adopted vessel monitoring systems (VMS) in order to track and monitor the activities of fishing vessels. The VMS approach is not flawless and as such, there are still known cases of IUU fishing. The present work is integrated in a project PT2020 SeeItAll of the company Xsealence and was included in INOV tasks in which a monitoring system using video cameras in the Ports (Non-boarded System) was developed, in order to detect registrations of vessels. This system registers the time of entry or exit of the vessel in the port. A second system (Boarded System) works with a camera placed in each vessel and an automatic learning algorithm detects and records fishing activities, for a comparison with the vessel's fishing report.

Keywords — Fishing, Vessel, CCTV, Image Processing, Plate Recognition, Machine Learning, Computer Vision, Machine Vision

Resumo — A pesca ilegal, não declarada e não regulamentada (INDNR) é uma ameaça global tanto para os ecossistemas marinhos quanto para a pesca sustentável. Os produtos INDNR são frequentemente provenientes de pescas que não possuem medidas de conservação e de gestão, o que permite a violação dos limites das capturas ou a captura não declarada. Para contrariar esse problema, alguns países adotaram sistemas de monitoramento de embarcações (VMS) para acompanhar e monitorar as atividades dos navios de pesca. A abordagem VMS não é perfeita e, como tal, ainda há casos conhecidos de pesca INDNR. O presente trabalho encontra-se integrado num projeto PT2020 SeeItAll da empresa Xsealence. Este trabalho integrado nas tarefas do INOV no qual foi desenvolvido um sistema de monitorização das entradas dos navios nos Portos (Sistema não embarcado) no qual pretende-se desenvolver um sistema que detete as matriculas dos navios registando a hora de entrada e saída do porto com recurso da camaras de vídeo. A outra componente (sistema embarcado) é colocada em cada embarcação uma camara de video e, recorrendo a aprendizagem automática e um sistema de CCTV, são detetadas as atividades de pesca e gravadas, para posterior comparação com o relatório de pesca do navio.

Palavras-chave — Pesca, Embarcação, CCTV, Processamento de Imagem, Reconhecimento de matriculas, aprendizagem automática, visão artificial

Acknowledgments

Firstly, I would like to express my sincere gratitude to my advisor Professor João Ferreira for the continuous support during this dissertation, for his patience, motivation, and experience. His guidance was crucial to accomplish this dissertation. I would also like to thank my girlfriend, my family and friends for all the support.

I. INTRODUCTION

A. Motivation and framing

Illegal, unreported and unregulated fishing (IUU) is one of the greatest threats to the preservation of endangered species, responsible for the destruction of marine habitats, distortion of competitive logic, weakening coastal communities and places licensed fishers at a disadvantage. According to data from the European Commission of fisheries and maritime affairs (European Commission, 2016a), illegal fishing accounts for about 15% of global catches, amounting around € 10 billion / year.

The EU, together with other international organizations, is making efforts to address the gaps that allow illegal operators to profit from their activities. Among the issues to be considered, the European Commission of fisheries and maritime affairs (2016) highlights the following:

- The catch report does not have a direct inspection, which means the reported information is based exclusively on the information given by the master / captain of the vessel;
- The control regulation only provides for the observer (on-board inspector), which is only used in specific international control areas;
- The only point of control is the landing, where the species and weights are recorded; There is no control over discards at sea.

Fisheries control focus not only on the verification of fishing activities at sea, but also acting on all points of the chain, since the time of leaving the vessel to the placement of fish on the market. The management of the fishing activity is based on obtaining estimates of fish abundance and mortality imposed by fishing.

In order to promote the sustainable exploitation of marine resources, a European quota system has been set up to manage the catch of various species of fish according to various parameters such as abundance, reproductive cycles and their economic value. However, limiting does not reduce the total catches as fishermen optimize the use of their catching capacity by discarding low-value fish, and there is currently no way to quantify discarded fish. In addition to the fish disposal, there is sometimes a deviation between the fish caught and the fish declared by the vessel that overflowed it before reaching the port.

Most of the existing technologies allow the monitoring of vessels during their activity, tracking the vessel's position over time and determining the fishing activity through logbooks or other reports produced by the vessel's captain. As such, currently, the only reliable means of verifying the fishing activity are the on-site inspection from the exit of the port until the return of the vessel.

B. Objectives

The proposed system aims to provide the surveillance authorities with an automatic solution that responds to the needs identified in the monitoring, control, management and surveillance systems of fishing activities. To meet those needs, it is intended to develop a surveillance system that allows the control of entrances and exits of fishing vessels in the port, in particular by collecting and processing vessel profile images that identify call sign, IMO or registration. A database holds a log of the port's entries and exits by associating the ship's identifications with the detection timestamp.

The approach followed is based on a CCTV (closed-circuit television) system, therefore it needs support from image processing to recognize objects.

In parallel with the inspection at the port, the project also includes a second module composed of a Vessel Monitoring System (VMS) (European Commission, 2016b), also equipped with a camera, both placed on the vessels. The devices determine the characteristics of the fishing techniques being used in the vessel. The goal is to analyse whether or not there existed a moment during the journey where the crew caught fish, which method was used and the duration of the catching. In addition, the VMS unit records the location of the vessel. The implementation of both the hardware and the VMS unit is outside the scope of this thesis.

The data captured by the cameras of both modules is evaluated through a solution based on machine learning techniques, capable of predicting various forms of the solution and evolving based on the number of cases. In this sense, it is intended to make use, when possible, of rule models from the capture of images through fixed cameras and/or amateur videos, in order to train the algorithm to find the boats and the corresponding identifications, for the first module, and the gear/methods used, for the second module.

The automation of the identification and recognition is one of the most difficult challenges of this work, especially the recognition of the characters of the registration, since it is intended to obtain correct data with a great success rate, without consuming too much time in the analysis, compromising the authorities' actuation.

C. Research questions

The main goal of this work is to answer the following question:

- "How to create a system capable of interpreting the license/registration of a fishing vessel and identifying the gear/methods of the boats in a robust way, attending to similar technologies already developed?"

In order to answer this question and due to its degree of complexity, there is a need to adopt the methodology of dividing and conquering, that is, to divide this question into smaller questions so that they can be answered easily and directly. The division consists of two main systems: the registration retrieval and the identification of the fishing gear.

Regarding the registration retrieval, some obstacles must be dealt with, including non-contrasting colours to display the registration, vague standards in respect to font types and sizes, lack of specifications on the registration's positioning and distortion. The following questions stand out:

- "Which are the best methods/algorithms to find fishing vessels, attending the need of satisfactory performance in adverse situations (e.g. bad weather and occlusion)?"
- "Which methods have a performance that enables responses to be delivered as quickly as possible and with an acceptable level of successful recognition?"

As for the identification of the fishing gear/methods, similar issues regarding standards occur, as such the following questions are raised:

- "Can the previous algorithms be applied to find the vessel's registration number? If not, which are the most appropriate methods for this purpose? "
- "What is the most robust method (s) for this module in relation to a probable set of limitations that may have similarities and differences relative to the other module (occlusion, illumination, atmospheric conditions, etc.)?"

D. Investigation method

Concerning the research methodology followed in this thesis, the Design Science Research is a problem-solving process and an information technology research methodology that focuses on the study, development and performance of innovative artefacts containing knowledge. We'll be focusing on the contributions of Peffers et al. (2006-8), Hevner et al. (2004) and Cardoso (2001). With this methodology, we ought to create new artefacts contributing to the construction and evaluation of generic means-ends relations. Design Science Research reveals the knowledge and understanding of a design problem and its solution are acquired during the building and application of an artefact.

This thesis is composed by the following artefacts, corresponding to each part of the system:

- Non-Boarded system:
  - A module (composed by a computer vision algorithm) capable of identifying moving vessels and determining the direction of the motion (entering or exiting the harbour).
  - A module (composed by a computer vision algorithm) capable of recognizing most of the regions containing characters present in the vessel.
  - A module (composed by an OCR algorithm) capable of interpreting the regions determined in the previous module and obtain a portion or totality of the characters.
- Boarded system:
  - A module, composed by one or several algorithms, able of detecting, tracking and counting the fishing gear in the ship.

This procedure has several components designed in a sequence, that is the output of the previous state is the input of the next, while maintaining the possibility of iteration, retroceding to the faulty component. The Design Science Research suggest 7 guidelines for information system research:

1. **Design as an artefact** – the goal is to produce a viable artefact in the form of a construct, model, method or instantiation. In this work, the artefacts we aim to develop are a model that combines state-of-the-art methods to accomplish two goals: identify fishing vessels nearing a harbour and detect when a fishing occurs observing the behaviour of the fishing gear.
2. **Design as an artefact** – the goal is to produce a viable artefact in the form of a construct, model, method or instantiation. In this work, the artefacts we aim to develop are a model that combines state-of-the-art methods to accomplish two goals: identify fishing vessels nearing a harbour and detect when a fishing occurs observing the behaviour of the fishing gear.
3. **Problem relevance** – which states the goal of Design Science Research is to develop technology-based solutions to matters of relevance. The introductory section introduces the importance of this thesis.
4. **Design evaluation** – This methodology requires the demonstration of quality, rigour and efficiency of the artefacts through well-designed test cases.
5. **Research contributions** – Design Science Research also suggest the clear and verifiable contributions in the design phases, that serves the purpose of both avoid plagiarism and inefficiency caused by unknown best fitted alternatives.

6. **Research Rigor** – Rigorous methods in both construction and evaluation of the design artefact are advised.
7. **Design as a search process** – The effective use of available means to reach the desired end while satisfying the laws in the problem environment results in an also effective artefact.
8. **Communication of research** – The presentation must be well-presented for both technology and management-oriented audiences.

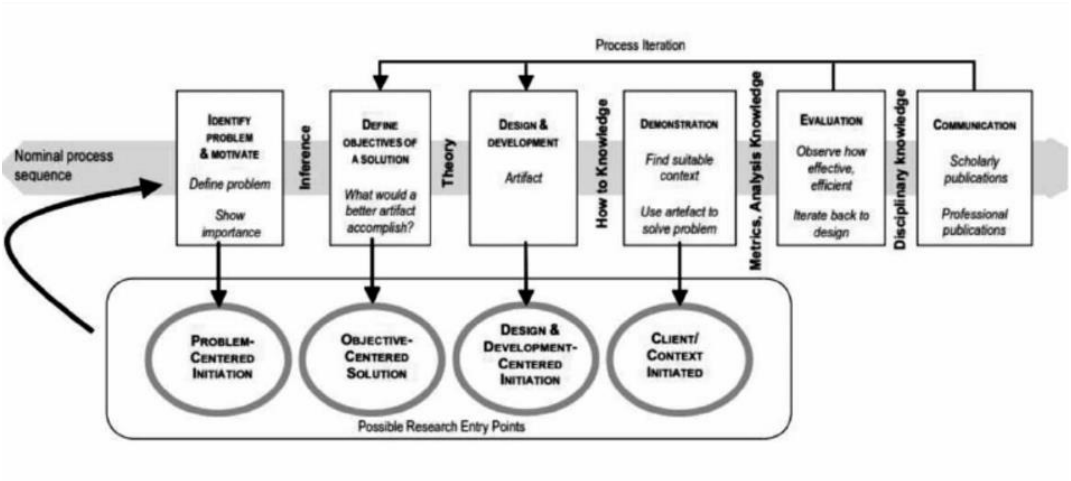


Figure 1– Design Science Research process model according to Peffers et al. (2007-8).

In parallel to the guidelines, this methodology (Figure 1) is also composed by 6 main activities:

1. **Identification and definition of the research problem** – the model suggests starting with the identification and definition of the research problem as well as its relevance, which contributes to the development of an artefact capable of supplying a solution. It is necessary to include knowledge of the state of the art of the problem. This activity is contained throughout the introduction of this thesis and later extended in the literature review section, where it is explained the importance of a system capable of identifying fishing vessels and check the conformance of the fishing report by analysing the VMS data, or in other words, whether the fishing gear was used or not.
2. **Define the objectives/requirements of the solution** - The next step is the definition of the objectives/requirements of the solution, determining what is conceivable either quantitatively or qualitative, followed by the design and development of artefacts that may explain specific segments of the problem. The introduction and the methods discuss and develop this topic in detail, but, to put it in short, and recalling the research questions, it is required an image processing

algorithm, or a combination of image processing algorithms, to be run in a real-time scenario, with the highest detection rate within possible, attending they must ultimately achieve their goals in a great variety of scenarios: identify the fishing gear and determine if any fishing activity is occurring, in the case of the boarded system; and identify a fishing vessel entering or exiting the harbour, in the non-boarded system.

3. **Design and development** - The third phase involves specifying the desired functionality of the artefact and its creation. In this phase, the resources necessary are all the knowledge that can sustain a possible solution. The methods section details more about this topic, explaining and detailing the conditions in which the system/algorithms must be able to work, which scenarios are optional or non-relevant, the conditions in which the system can be called a success or a failure and relevant details about the development of this artefacts.
4. **Demonstration of the artefact** - The succeeding step is the demonstration of the artefact and the goal is to prove its effectiveness to solve ate least one of the instantiations of the problem. It is essential to measure how the objectives of a solution match the observed results from the use of the artefact. As such, in this thesis, the solution must answer the research questions therefore fulfilling the core functionality, that is to detect and help identifying fishing vessels near the harbour and detecting when a vessel is fishing. At this moment, little to no progress has been made, therefore no relevant information was placed in this thesis.
5. **Evaluation of the artefact** - The fifth stage is a further evaluation and can either be a comparison of the artefact's functionality with the solution's goals or a quantitative performance measure. In this work, the solutions are evaluated attending to its performance in the detection and identification, along with its processing time. The system was tested initially in a simulated scenario using a programmable and dynamic animation, simulating with higher or lesser details, and later in a real-life scenario and, if possible, with real fishing vessels.
6. **Communication** - The final phase is the communication of the research developed. The usefulness of the artefact is announced to relevant audiences and professionals. This is the purpose of the papers that was released during the realization of this thesis.

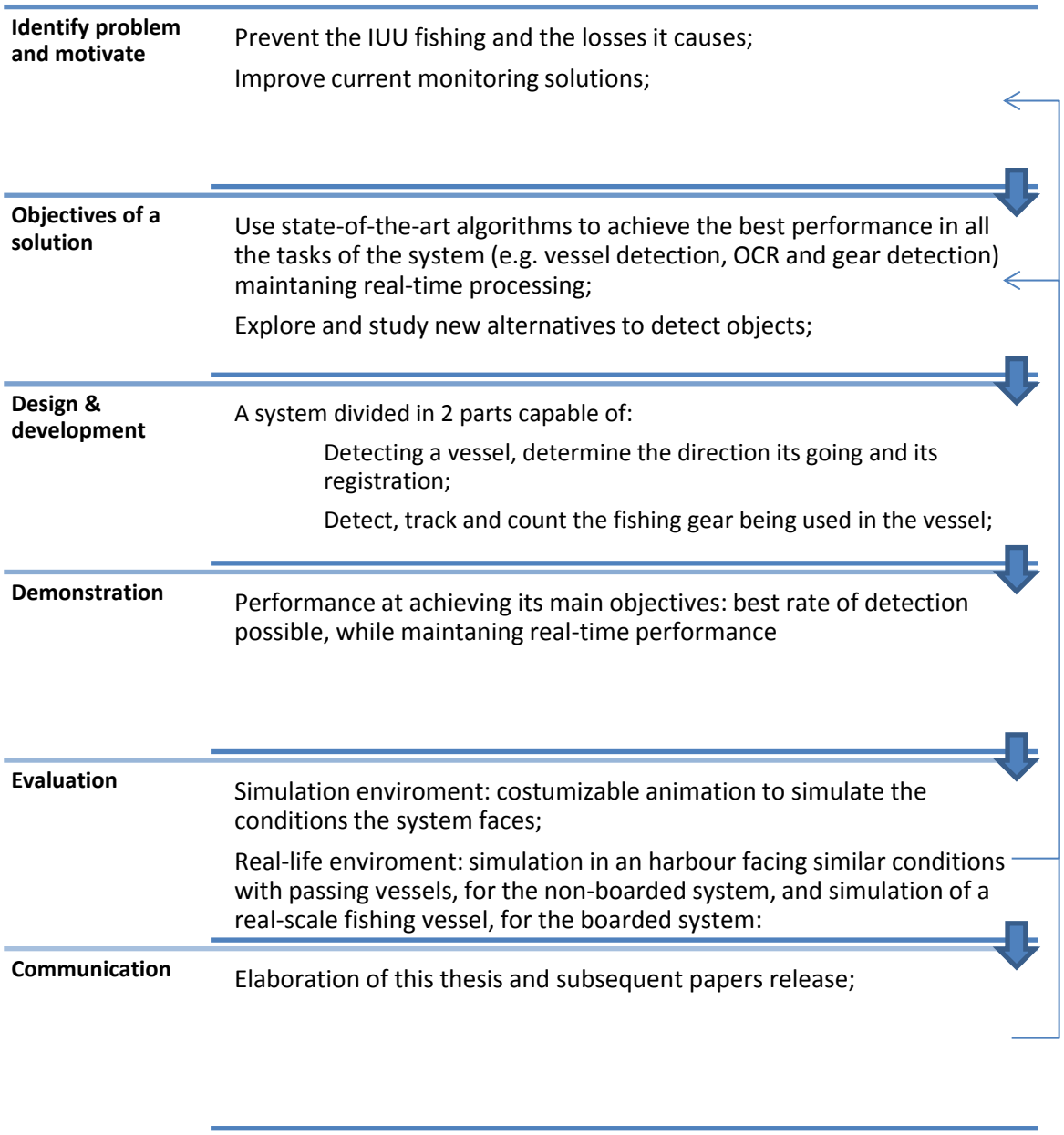


Figure 2 – Design Science Research process model (Peffers et al. 2007-8) adapted to this thesis

The DSR model suggest 4 major approaches:

1. **Problem-centred approach** – This approach is starts from the first activity resulting from the observation of the problem / suggested research.
2. **Objective-centred approach** – On the other hand the objective-centred approach starts on the second step (define objectives) and typically comes from industry and/or a research need and involves creating an artefact to address the problem.
3. **Design and development-centred approach** – Starts with the third activity (design and development) and is the suitable approach for an

already existing but still incomplete artefact. This approach focuses on adapting the artefact to fit as a solution for the specific problem.

4. **Client/context initiated solution** – Is generally based on the observation of a previously implemented solution that worked. The goal is to replicate previous activities and apply rigour to the research.

The problem was previously identified and studied by the European Commission of fisheries and maritime affairs and later by both INOV INESC INOVAÇÃO and Xsealence. Considering this research started with an already mature identification and definition of the problem, this research follows the Objective-centred approach, starting with the definition of the objectives and specification of all the conditions in which the algorithm must work. The remaining steps are all part of this research and correspond to the elaboration of the artefacts, demonstration and evaluation.

E. Contributions

The research made during this dissertation originated the following scientific contributions:

- Ferreira, J. C., Branquinho, J., Ferreira, P. C., & Piedade, F. (2017, June). Computer Vision Algorithms Fishing Vessel Monitoring—Identification of Vessel Plate Number. In *International Symposium on Ambient Intelligence* (pp. 9-17). Springer, Cham.

II. LITERATURE REVIEW AND RELATED WORK

A. Overview

There are several different ways to monitor commercial fishing operations. The traditional approach includes undertaking dockside compliance and fish market visits. Other approaches have been tackled by numerous authors (McElderry, 2006; Witt & Godley & Christensen & Guenette & Pitcher & Mountain, 2007; Jennings & Lee, 2012), including: using aircraft (manned or unmanned) to overfly fishing vessels; using patrol vessels to undertake at sea boarding or surveillance; using VMS that use satellite positioning data to work out the location and speed of the vessel; sending observers to sea for the duration of a sea trip to collect scientific data or evidence gathering for compliance; and using self-reported data, such as E-logs, paper logbooks, sale-notes, landing declarations.

Blaha described a system having similarities to the intended implementation (Blaha, 2016). This system uses image processing techniques with a VMS system attached to the fishing vessel. The VMS system is activated when it detects load, through weight sensors or when hydraulic mechanisms are activated. Once the VMS system is activated, the camera registers the moment of capture, analysing some of the fish caught in search of protected fish species. The data is stored locally and after each month is then submitted to the corresponding authorities for conformity check. On the other hand, the same system had low accuracy in detecting is private and therefore there isn't any specifications on the methods or equipment used.

In Mangi's work, in order to protect seabird populations attracted by fishing activities, an EMS (electronic monitoring system) approach was used to assist in compliance with fish catching standards (Mangi & Dolder & Catchpole & Rodmell & Rozarieux, 2013). Despite this type of systems typically are not exclusive to fishing vessels, the goal is similar, hence its use. The solution consists of a camera system that identifies the presence of birds caught in fishing nets, with a success rate of around 79%, proving that the use of EMS produces accurate results.

To summarize, there is quite a small number of solutions which try to address similar issues with image processing and computer vision. In order to get some starting guidelines, multiple computer vision applications were studied, and the best algorithms are discussed in this section. A common misconception is comparing this project with license plate detection algorithms designed for cars, since there are some considerable differences, including:

- **Contrasting colours** – Car license plates are of contrasting colours (black and white), while vessel registrations allow for a wide variation of colours, including non-contrasting colours. In addition, the white background in car plates is made of a highly reflecting material, while

the foreground is made of a non-reflecting material, easing the distinction of both colours.

- **Different font sizes** – there is a lack of standards in the registration size in vessel's, making it difficult to predict how large or small is the region we are looking for in the image, even knowing the distance to the vessel. The size often changes for each text, depending on its location in the hull and occasionally there is font size difference in the same word in order to adapt to the hull's characteristics.

- **Deformation** – car plates are flat, hence not deforming the text. The same can't be said about the vessel's registration which often are deformed by the hull's shape.

- **Text format** – while the license plate number commonly has a fixed format, composed of a fixed number of letters and a fixed number of numbers, some elements in the vessel do not have many formatting rules, such as the name, which can be composed of an undefined number of words of different sizes.

- **Exposed conditions** – Most license plate algorithms work if photos taken in a considerable controlled environment granted by the toll infrastructure, minimizing some atmospheric conditions such as rain and lack or excess lighting. This type of infrastructure may not be possible to provide in a harbour.

As previously mentioned there are 4 modules to implement, each having its own specification and can be solved by different image processing algorithms. Currently, some of the most popular methods used in image processing are visual descriptors, more specifically, frameworks such as HOG classifiers (Dalal & Triggs, 2005), Viola and Jones detectors (Jalled & Voronkov, 2016) or particle filters, mainly due to their performance, as well as training speed. Recently, convolutional neural networks also became quite popular, especially to solve image classification problems. This section reviews object recognition approaches, including some of those previously mentioned, together with their weaknesses and strengths, in order to answer the questions posed in the introductory section, using state-of-the-art technologies. The choice of the solutions is based on the fulfilment of the proposed objectives, valuing the solutions with the best rate of correct identifications, within a limited processing time, allowing the authorities to act if necessary. The purpose of this section is also to explore existing ways of tackling disadvantages if it proves to be rewarding. This sections also debates and evaluates some of the existing frameworks for computer vision and image processing.

B. Methodology in object detection

One of the most important steps it's the detection of objects: in order to find the vessel's plate, it is necessary to find the vessel in the first place; and to detect the presence of a fishing equipment it is also necessary to find the object. There are different methods to find objects but an adequate methodology is necessary to ensure robustness and performance.

A computer vision methodology varies depending on the needs, as well as the characteristics of the algorithm. Typically, each author separates each step in their own way, according to their needs and fitting the task.

According to Moeslund et al. (Moeslund & Hilton & Kruger & Sigal, 2011), the steps for a good performance and correct functioning of a computer vision algorithm should follow the following sequence (Figure 3): 1) detection of a new object; 2) classification of the same object into categories of interest; 3) continue to trace the same object while it is visible, thus avoiding repeating the first two steps. Keep in mind these authors develop this methodology for tracking objects. Justin Johnson and Andrej Karpathy (n.d.), use an architecture composed of: 1) Input, where N images of K classes are provided; 2) Learning, where the algorithms learn the features of each class of object; 3) evaluation were a new set of images is provided and algorithm tries to label each image. These authors are specifying this architecture to fit a convolutional neural network. On the other hand, the most commonly accepted model is a broken-down architecture, divided in 6 steps, as shown in Figure 3, which tends to be more general. All these methodologies were taken into account throughout this document. These approaches have different levels of abstractness, different goals, and some rely more on a machine learning type of approach more than other, but the same guideline is used in all three.

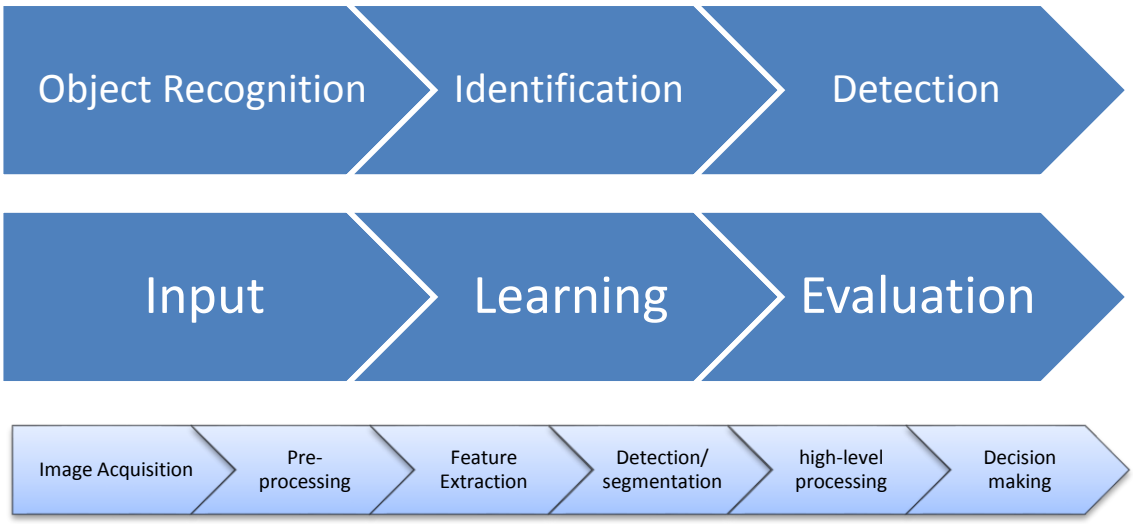


Figure 3 – Moeslund, Hilton, Kruger and Sigal's (2011) methodology for computer vision algorithms (above), Justin Johnson and Andrej Karpath's (n.d.)

methodology (middle) and traditional/most common methodology (bottom), according to the same authors (Justin Johnson and Andrej Karpath, n.d.).

There are mostly two main methods for object detection, that act as a starting point to all other algorithms and sub-methods: pixel-based detection and object-based detection. In essence, both methods use features to evaluate whether a pixel belongs to the object or the background, therefore there is a strong relation between feature detection and both pixel-based detection and object-based detection. Due to its importance a section below is dedicated to the detection of features, since it helps explaining some of the approaches later on. Additionally, these features can also be combined with other technologies, such as infrared cameras or complex artificial intelligence, giving rise to innovative high-performance alternatives, some of which are mentioned throughout this section (Moeslund & Hilton & Kruger & Sigal, 2011). This separation is sort of a “grey area”, since different authors have different views regarding grouping or separating these approaches, which use artificial intelligence or help from hardware, in the other two main approaches. In this context it makes sense to separate simpler approaches from complex approaches, therefore the infrared approach and the deep learning approach were separated from the remaining methods of detection.

C. Feature detection

Features define how a computer can identify a certain object, for example, it’s shape and colour. An algorithm searches for a set of rules in order to validate these features, plus a machine learning algorithm can even learn this features on its own. Features are necessary in order to find the objects we are looking for.

There is no universal definition of feature, but is considered a relevant part of an image, since they are what differentiates or associates objects, therefore features are on the basis of any computer vision / image processing algorithm (OpenCV, 2014). When comparing objects, we are looking for repeatability in features between two or more images. The concept of feature detection refers to methods able of making local decisions at every image point, and evaluate whether a given point contains a distinguishable image feature of a given type or not. The most common types of image features are:

- Edges – points where there is a boundary between two regions and typically have a strong gradient magnitude.
- Corners – refer to point-like features in an image and can be located by looking for high levels of curvature in the image gradient. It is also common to capture small bright points in darker backgrounds, which in the sense of the word is not considered a corner, hence this feature is also referred to as points of interest.

- Blobs – also called regions of interest, and often reference to areas in an image which are too smooth to be detected by a corner detector.
- Ridges – are considered elongated objects and can be thought of as one-dimensional curve that represents an axis of symmetry.

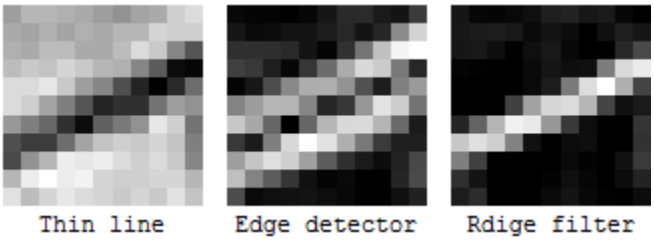


Figure 4 – Examples of some features and their differences (Opencv, 2014).

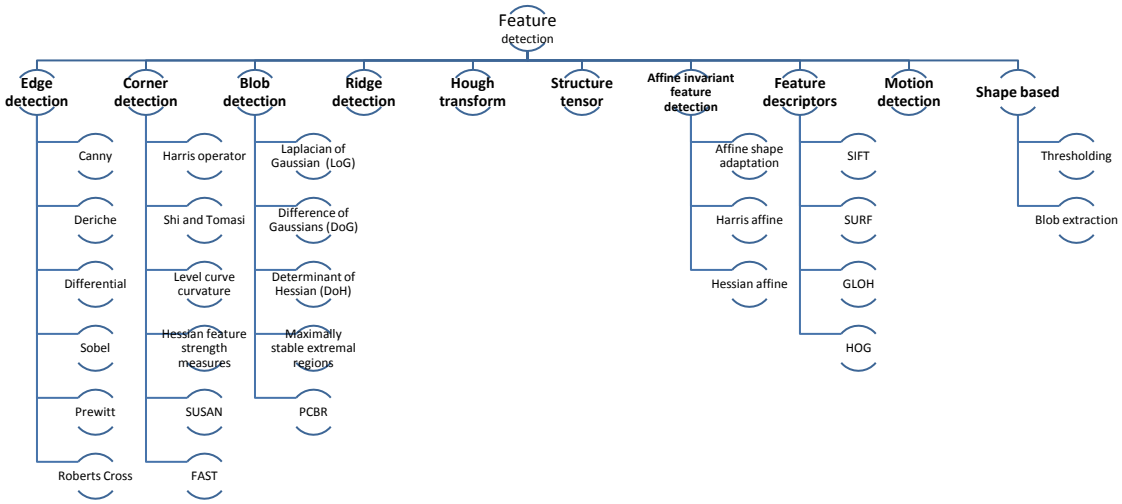


Figure 5 – Hierarchy of low-level algorithms in feature detection. These methods are used to extract features, which can later be used in pixel-based, object-based or hybrid detection algorithm (Moeslund & Hilton & Kruger & Sigal, 2011).

D. Pixel-based detection

In this model, the pixels of the image are extracted and compared with a previously developed model, in order to identify each pixel, either as background, or foreground (object). What distinguishes pixel-based detection from feature detection is that the first method parses the detected features and verifies if they match those of the object, while feature detection only focuses on finding features of an image. After performing this filtering, subtracting the background to the image, it is possible in theory to distinguish the silhouette of each object in the image. The pixel based detection method typically has better results in indoor environments where there are no significant changes in the background than in outdoor environments where changing brightness, shadows and atmospheric

conditions may be some of the sources of noise (Moeslund & Hilton & Kruger & Sigal, 2011).

This approach focuses heavily on finding objects that move or are introduced to the scene (Moeslund & Hilton & Kruger & Sigal, 2011; Heikkilä & Pietikäinen, 2006). It should also be complemented with filtering and blob analysis to distinguish the shapes of the object from any other non-relevant objects. For example, in the case of a given moving object, this technique can be used to remove other moving objects (cars, animals, etc.), as Haj et al. states (Haj & Fernández & Xiong & Huerta & Gonzàlez & Roca, 2013). Blob analysis uses the silhouettes to eliminate noise, but there is an added difficulty when there are shadows (Moeslund & Hilton & Kruger & Sigal, 2011; Perez, 2005), because they have similar traits relative to the object, thus requiring a reasoning about the context in order to eliminate the shadows (analysis of atmospheric conditions or colour comparison). This approach is typically lighter in terms of processing but typically has worse results as several authors agree (Lu & Tsechpenakis & Metaxas & Jensen & Kruse, 2005; Gao & Mas, 2008).

Figure 6 – Theoretical example of a process of pixel-based detection and subsequent segmentation, starting with the initial image (left), applying the most effective filter(s) (colour, edge, etc.) and obtaining a binary mapping of all the relevant pixels (middle). The result (right) is the combination of the mapping and the original image, and shows only the relevant information.

E. Object-based detection

Alternatively, in this second approach, a floating window of fixed size navigates through each frame of the video, scaling and undergoing translations in all possible locations of the image. A binary classifier evaluates the presence of the object for each position. The reason why the window is scaled relates to its ability to detect objects at different distances. Each relevant object found is marked with

a box / window that surrounds it, so that it is visible to a user. Methods such as Support vector machines (SVM), AdaBoost and Random Forest Classifiers have been successfully used in the past to sort the contents of the window (Moeslund & Hilton & Kruger & Sigal, 2011; Lienhart & Kuranov & Pisarevsky, 2003). However, to complete only one frame in its entirety it is necessary to adjust the window 10 to 100 thousand times, causing severe delays in the execution time. Using simple and sensitive classifiers, such as SVMs, to, in a first step, traverse the entire image and return a list of locations in the image where there is a greater probability of locating the intended object, while in a second step, a more effective classifier is run in the marked places, confirming the objects, is a good way to increase the performance (Mahadevan & Li & Bhalodia & Vasconcelos, 2010; Mehran & Oyama & Shah, 2009). This method is called cascading. This approach typically fails when the background is far from static, such as when a mobile camera is used for image capture, or when there is partial occlusion phenomenon. This approach typically uses HOG detectors constructed with the notion of edges produced by the object in certain well-defined and non-random locations. It can also be complemented by using Viola and Jones detectors, which combine simple features with cascade classifiers as well as contrast detection (Moeslund & Hilton & Kruger & Sigal, 2011; Lienhart & Kuranov & Pisarevsky, 2003). The Viola and Jones approach, like the cascade approach, requires a previously trained classifier with a set of positive examples and an even larger set of negative examples, so that proper extraction of the characteristics occurs. Precautions should also be taken to avoid overtraining (Lienhart & Kuranov & Pisarevsky, 2003). This approach presents better results, especially in outdoor areas with the possibility of partial occlusion (Moeslund & Hilton & Kruger & Sigal, 2011).

Figure 7 – Theoretical example of an object-based detection algorithm, starting with the original image (left), applying a previously trained cascade classifier (middle) and the resulting window with the relevant object

F. Infrared-based detection

The authors in (Abuarafah & Khozium & AbdRabou, 2012) demonstrate that when there is a large number of objects in the same image, making a real-time identification becomes inefficient or even impossible. In this context, the author proposes the use of a forward looking infrared (FLIR) camera to monitor and estimate the density of people, given the maximum capacity of the site. In the case of the fishing detection module, the presence of heat means the presence of fishermen or fish, which in a significant amount evidences that the vessel is fishing. The use of infrared is based on the fact that at an ambient temperature of 300 Kelvin, a body emits wavelengths in the order of 9.7 μm, situated in the middle of an interval that makes them appear bright. This approach also allows you to obtain images even without the presence of natural light. After the image capture, a geometric algorithm is used to analyse the silhouettes and reduce the number of false positives (for example, motors) (Abuarafah & Khozium & AbdRabou, 2012). This method is not completely foolproof, since in closed or very hot environments it is difficult to detect the differences at thermal level, as well as very bright places for the same reason, as evidenced in (Abuarafah & Khozium & AbdRabou, 2012). This method also requires a calibration taking into account the context, since the human temperature varies in the FLIR cameras according to the time of capture (whether it is day or night) and surrounding temperature. According to (Moeslund & Hilton & Kruger & Sigal, 2011), this calibration must occur every 30 minutes.

Figure 8 – Example of an infrared picture containing people. As shown in the Figure 8, colder surfaces are represented with darker colours and hotter surfaces, like skin, are represented with lighter colours (Anon., 2017, available at: https://alfa-img.com/show/thermal-infrared-people.html).

G. Part-based detection

There are hybrid methods that use both pixel-based, as well as learned features to detect objects: holistic representations (Moeslund & Hilton & Kruger

& Sigal, 2011) (Mahadevan & Li & Bhalodia & Vasconcelos, 2010; Mehran & Oyama & Shah, 2009), part-based (Moeslund & Hilton & Kruger & Sigal, 2011) and local characteristics (Moeslund & Hilton & Kruger & Sigal, 2011; Lienhart & Kuranov & Pisarevsky, 2003). These methods generally are not the most suitable to carry out this type of analysis, since they require very detailed images, long processing times, are sensitive to dynamic scenarios and do not perform well in the event of occlusion (e.g. tires covering the inscription or waves/water). They may, on the other hand, facilitate the task of finding the vessel and the direction in.

H. Convolutional neural networks

Recently, with advances in neural networks, neural convolutional neural networks (CNN) and deep neural networks (DCNN) have gained great popularity, thanks to their efficiency and speed in detecting objects. This approach, because it supports video filmed on dynamic cameras, is also used in some self-driving cars (Perez, 2005). Its disadvantages are the speed, since the processing can even reach the order of minutes per frame, needing task-oriented or optimized hardware (Gao & Mas, 2008), and the fact that, like any neural network, its performance depends on the training of the network (Moeslund & Hilton & Kruger & Sigal, 2011; Gao & Mas, 2008), which is a complicated and time consuming process. For the correct functioning of a neural network, proper training is necessary so that the network can identify objects and distinguish them from the remaining objects in the scene, not forgetting that, like classifiers, too much training can cause the network to recognize only the instances of the Objects used for training, rather than a particular class of objects (Simard & Steinkraus & Platt, n.d.). Other factors to consider are the minimum of acceptable accuracy and the maximum acceptable processing time, since increasing precision increases processing time and vice versa. Angelova et al (Angelova & Krizhevsky & View & Vanhoucke & Ogale & Ferguson, 2015) state that they have developed a network, composed of a set of DCNNs arranged in cascade, with a precision rate of 75%, capable of analysing 15 frames in a second, quite revealing results for the project to be developed, where the rapid achievement of results is valued.

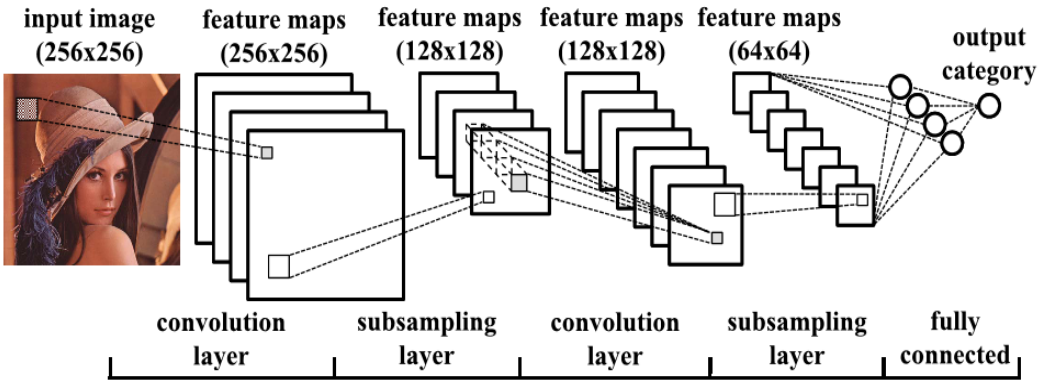


Figure 9 – Example of the mechanisms behind a convolutional neural network (Opencv, 2011-4)

I. Frameworks

There are not many frameworks dedicated to computer vision/image processing and even a fewer number of open source frameworks. OpenCV (Opencv, 2011-4) is by far the most popular tool and also the one with more variety of support platforms – Android, iOS, Linux, Mac OS and Windows – and also programming languages – C/C++, Python and Java. OpenCV is a library of programming functions aiming for real-time applications with great efficiency, displaying optimized libraries, which can take advantage of multi-core processing. This framework is widely accepted worldwide and its community reaches 47 thousand users in diverse areas. Some software tools such as Matlab use OpenCV libraries for computer vision and image processing purposes (Matlab, 1994-2017).

Alternative tools such as SimpleCV tend to base themselves on OpenCV and end up using some of them (SimpleCV, n.d.). PyCVF (Python Computer Vision Framework) is a Python only alternative, but has certain limitations both on the support and compatibility (Python Computer Vision Framework. n.d.). FastCV is a framework available only for mobile and focuses on gesture recognition, face detection, tracking and recognition, text recognition and tracking and augmented reality (FastCV Computer Vision SDK - Qualcomm Developer Network, (n.d.). Xpcv (n.d.) is another modular cross-platform framework designed for rapid prototyping of computer vision systems and its programming works on a drag and drop interface with blocks of code, in which an input is received and an output is transmitted to the next block. Each block can be configured with different parameters. Khoros (n.d.) is an older alternative from 1995 with few supported systems.

The most fitted framework for the purpose of this thesis is OpenCV considering its open source, support in both the operating systems level as well as diversity

of compatible programming languages and for its large community, which can provide useful suggestions.

III. METHODS AND SYSTEM DEVELOPED

A. System architecture and constraints

This section details the system architecture, using a top-down approach, starting from the solution as a whole, and then breaking it down to two parts and again breaking each of them in their corresponding modules. In this chapter, when analysing each component, all the constraints are detailed. Each component is also compared with different state-of-art methods and algorithms.

The proposed architecture is composed of two parts, as previously mentioned: One of the parts registers the identification of the fishing vessels entering or exiting the harbour (non-boarded system) and the other part monitors fishing activities in each vessel (boarded system). These parts of the solution are addressed as systems. Figure 10 clarifies how both systems fit in the overall architecture.

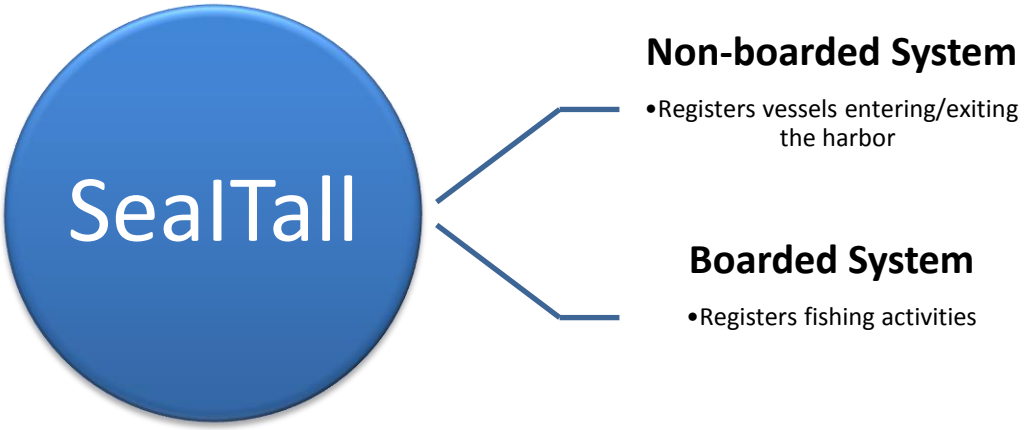


Figure 10 – The solution and corresponding parts.

Recalling the previous sections, some of the algorithms discussed are eliminated as they are not fitted for the task: The convolutional neural networks may require too much time, data and better hardware, which are all resources unavailable during the elaboration of this project; Due to hardware and budget limitations it is not possible to install any infrared cameras in the boarded system, despite the infrared could be used to find humans and determining, by their actions, if the kind of activity they were doing was related to fishing. This solution is also not feasible in the non-boarded system considering it would not contribute to identifying vessels.

The implementation of both systems occurred in two main phases: phase of experimentation and testing of algorithms in order to identify the best solution and the phase of implementation of the algorithm into their respective platforms: a computer for the non-boarded system and a VMS unit for the boarded system.

The first phase (testing) was performed using MATLAB because it is a popular high-level language in the area of computer vision, reducing effort and implementation time. The system created was called Test Environment and the idea is to quickly understand the capabilities and performance of a given algorithm and evaluate its implementation effort. There is, however, a disadvantage: by the ease and optimization of MATLAB it is common for complex applications to be developed, making it difficult or even impossible to implement it in another language, which can lead to the redesign of the solution.

The second phase, Production Environment, is different for each of the systems. As for the Non-boarded system, it consists of the implementation in .NET with access to DLLs developed in C ++, containing the algorithms tested in the Test Environment. The .NET platform is used due to the ease of implementation with the hardware component and its robustness, being essentially in charge of managing the hardware and coordinating it with the algorithm. Figure 12 illustrates how the components are connected in the production environment. Despite the existence of wrappers, the algorithm was developed in C++, in order to maintain all OpenCV functions without having an abstraction of variables that may have led to non-linear behaviour. The development was iterative and sometimes it became necessary to go back to the first phase to redesign the algorithm when converting from MATLAB to C ++ due to its complexity. Figure 11 outlines the tasks of each of the environments.

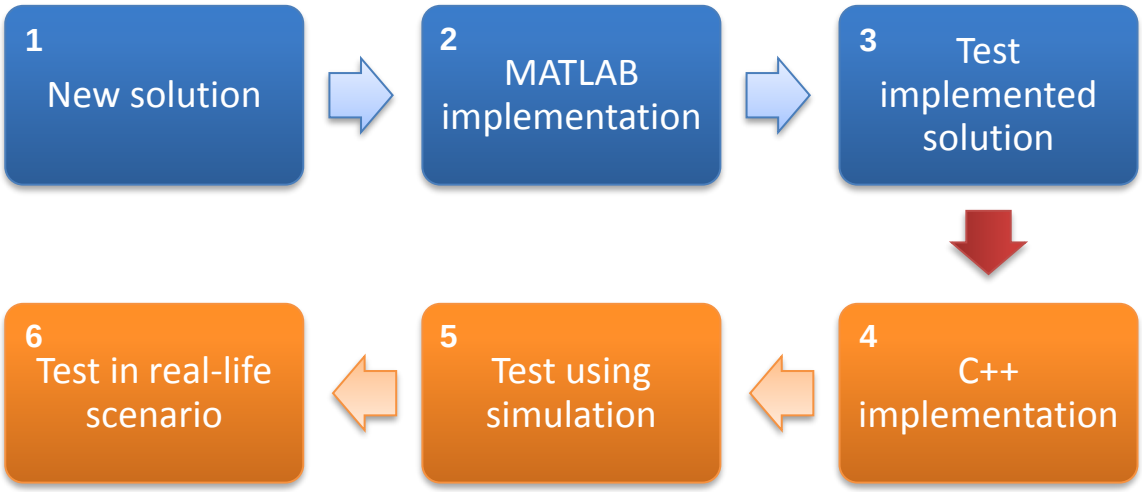


Figure 11 – The test environment is represented in blue, corresponding to steps 1 to 3. The production environment is represented in orange.

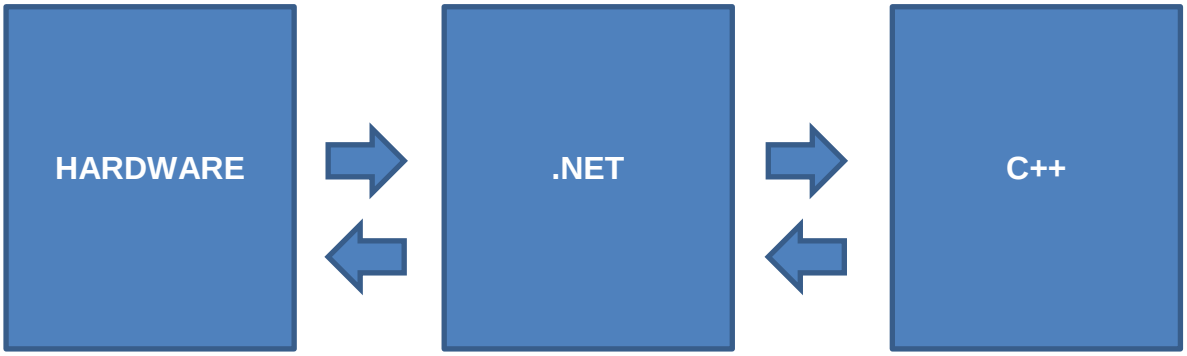


Figure 12 – Abstraction of the roles of each component in the production environment.

The boarded system's production environment is design in a similar fashion as the previous system, with the exception the developed processing unit is now as shared object instead of a DLL, because this system's target operating system is Linux, instead of Windows. Also, the .NET platform is a black box controlled by Xsealence and can be changed without affecting the algorithm's performance.

B. Non-boarded System

As for the non-boarded system, it determines vessel's registration by analysing the photo taken by the photographic camera, which is triggered when a video (context) camera detects a vessel passing by. As previously explained the algorithm is written in C++ and is managed by a .NET executable. The algorithm is composed of three modules. Despite being in the same DLL, each module is called separately and report to the .NET executable, which may or not activate the next module. The three modules are:

- **Vessel detection module -** responsible for finding a possible vessel in the surrounding area, using the context camera, and trigger the photographic camera;

- **ROI (region of interest) analysis module -** which analyses the high-resolution image for regions that may contain elements of interest (namely areas containing characters);

- **OCR module -** tasked with recognising the text in each area detected by the previous module. Additionally, the results are stored locally and can be later retrieved using a web service.

Figure 13 sums up the constitution of the system, including relevant hardware components and the three software modules. In this diagram, there is no distinction between the DLL and the executable. The lighter colours are used to represent the software and the darker colours the hardware. At the bottom, there is an example of input and output of the system.

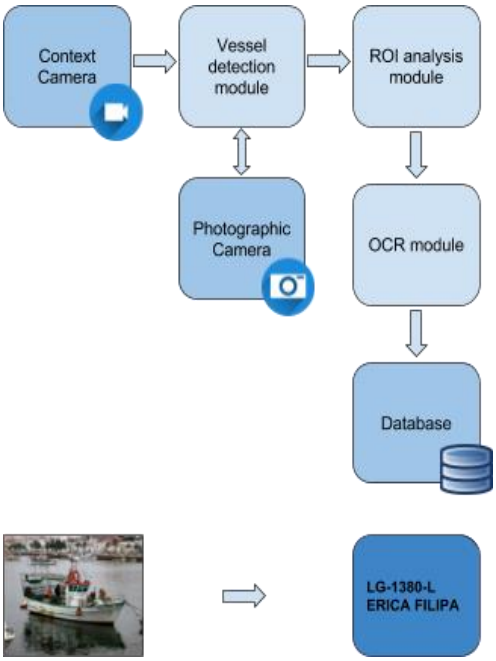


Figure 13 - Diagram of the relevant hardware and software that compose the non-boarded system (top), and example of a possible input and expected output (bottom).

The non-boarded system is operated with high resolution images, therefore eliminating the maximum noise possible is imperative. One simple and effective approach is to apply a filter to the colours of the image, more specifically converting the whole image to monochromatic and then eliminate all values that are neither light or dark.

The non-boarded system was placed in a harbour (Nazaré's harbour) and initially was considered two options regarding the placement of the camera, as shown in Figure 14:

- Option 1 – Entrance closer to the sea:
  - Allows a 100-meter gap between each side of the canal;
  - Public access;
  - Possibility of contemplating undulation.
- Option 2 – Entrance near the core of the harbour:
  - Passage width of 80 meters;
  - Restricted access with lower chance of vandalism;
  - Protection against undulation;
  - Vessels slow down near the narrow passage.

As different locations may require different algorithms, considering the inputs are disparate as well, the location chosen may be relevant in order to guarantee correct results. Option 2 proved to be the most advantageous as are discussed in the rest of this section.

Figure 14 - Geographic characteristics of the harbour and the two possible camera positions (Google Maps, 2016)

The hardware concerning the non-boarded system is composed of two cameras: one low resolution video camera (Bosch's DINION IP starlight 7000 HD) and a photographic high-resolution camera (IDS' UI-5490RE-C-HQ). The filming camera seeks moving vessels nearing the canal, triggering the photographic camera when the ship enters or exits the harbour, as such this camera is addressed as context camera. The choice of the cameras took into account a previous study where the requirements to ensure a minimally satisfactory operation of the solution for the site were analysed, considering a limited budget for the project.

In this way, for the reasons previously highlighted, the passage with 80 meters of width was chosen. It should be noted the depth of the passage is deeper in the centre and not navigable at the edges, giving a margin of about 10 to 15 meters from the edges was considered. In cooperation with several agencies selling CCTV equipment, the chosen video camera has a better price-quality ratio, guaranteeing an ideal quality between 24 and 70 meters, for a lens aperture of 29.91º, an exposure time of 3.33 and considering the maximum navigable speed of 10km / h within the channel, covering most of the channel's navigable area. Taking into account the camera's settings, the resolution is 10.6 Mpix, producing an image with 10 to 30 pixels per letter, and ensuring that letters up to 9 meters' high are still fully framed and readable in the image.

For the photographic camera, a maximum stern width of about 4.5 meters, a maximum stern length of 5.5 meters and a camera height of 5 meters was considered. With the camera parameterized for a lens aperture of 28.1 ° and a resolution of 720pixels, a CCD ratio of 9, a slope of 36.3 ° and considering a loss of 1 meter resulting from a dead angle of 11.3 °, the visible length of Stern is

about 5.47 meters. The vision of the camera covers a width of 4.76 meters next to the camera and 9.71 in the maximum length of the stern.

a. *Comparison with similar systems*

A common fallacy during the project design was the comparison with car registrations. There are a number of differentiating features that can make it difficult to apply algorithms used to detect car plates. Many of these differences relate to existing standards or, generally, to their lack, essentially in the construction and maintenance of vessel registrations. The understanding of these differences is critical to the understanding of the choice of algorithms and approaches used, since due to the purpose of the objects to be found and the human being's ability to identify characters, we easily assume similarities that may not be linear for an image processing algorithm. These differences are described below in Table 1.

Feature	Cars	Vessels	Relevance
Bounding box	Yes	Optional	Ease of delimiting and extracting straight lines and contours
Standard colors	Yes (black letters in white background)	No	Ease to search for certain colors, by intensity or values (Hue or RGB)
Contrasting colors	Yes	Yes, but open to interpretation	Ease of identifying contours and intensity changes
Standard font size	Yes	No, but with maximum and minimum limits	Ease of calculating the angle of skewness and rotation and ease to search for contours with the same thickness
Occlusion, degradation and/or damage	Unusual and usually mild	Common and more severe	Phenomena of occlusion, rust or damage may prevent the correct identification of one or more characters, as they may partially or completely hide

			characters from the registration
Control of the environment	Controlled (toll)	Little control (exterior)	A controlled environment that limits angles, speeds, camera-to-vehicle spacing, lighting and environmental phenomena (e.g. rain or fog) allows faster and more efficient processing
Skewness and rotation of license plate characters	Uniform between vehicles and between the characters of the license plate	Varies depending on hull position and curvature	A large variation of the skew between the characters of the same registration may prevent the calculation of a slope angle that satisfies all the characters

Table 1 – Comparison between some of the characteristics of car and boat registrations

Table 2 shows examples of car registrations and vessel registrations, as well as their representation after a thresholding operation using the Otsu method (Otsu, 1979), using 16 bins. In Table 3 are other common methods for license plate detection using pixel-based detection. Starting from the thresholding operation, it is possible to observe that some of the letters of the boat registration are not detached from the shade caused by the hull. Note the presence of the well-defined bounding box on the car registration plate and the corresponding line detection (Table 3, line 2). From the orientation of the detected lines it is possible to calculate the skew of the image. In the image of the vessel, even with the unusual edges around the characters, it is possible to notice the noise caused by shadows and the curvature of the hull, making it difficult to determine the skew using these lines. This problem is compounded by the fact that the letters of the vessel can still be painted with a light coloured on a dark background. This lead, in addition to determining the colours of the letters, to the inversion of the colours of the binary image. After identifying and filtering MSER features on both images, it is now possible to obtain the car registration letters along with the car symbol.

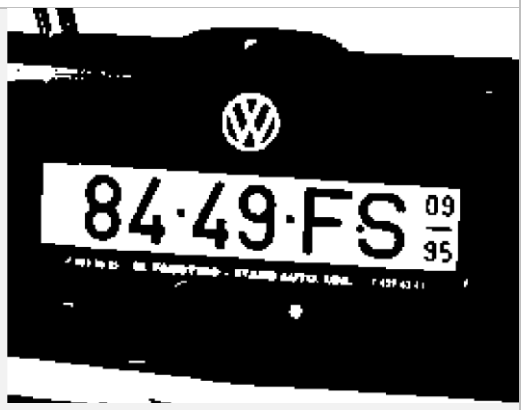
Original	Binary Image
	
	

Table 2 – Results of the application of the Otsu's binarization method

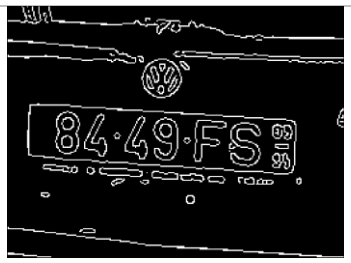
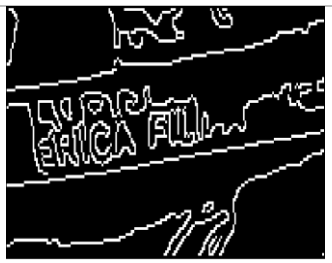
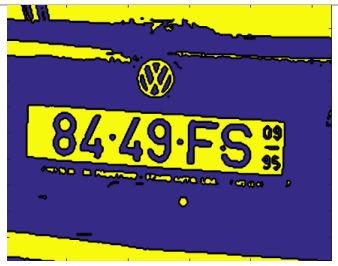
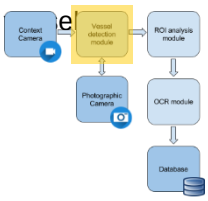
Contour detection using Canny's method	
	
Line detection using Hough's transforms	
	
Contour clustering by intensity	
	
Identification and filtering of MSER features for the determination of zones containing characters (the color in the image)	
	

Table 3 – Application of common algorithms in the license plate recognition to a registration of a vessel



b. *Vessel Detection Module*

For the detection of passage of vessels, we chose simpler approaches, so as to allow rapid action by the software. In other words, it is intended to take high-quality photography when the vessel is navigating in front of the camera, so the processing delay is a variable in consideration. The chosen approach should be lightweight in order to allow the camera to be activated at the right time. In this way, the pixel-based detection approach is the most appropriate since the processing times are considerably lower and because a greater number of false positives are preferred. The false positives can later be eliminated in the other modules. A greater number of false positives is preferred to missed vessels because the failure to identify a moving vessel compromises the performance of the entire system. Figure 15 describes how the field of view of the cameras is positioned.

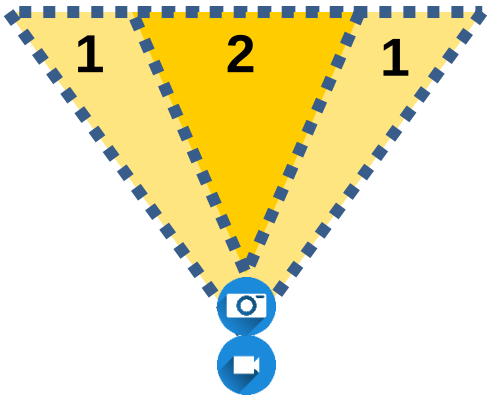


Figure 15 – Field of view of the context camera (1) and Field of view of the photographic camera (2)

The field of view of the photographic camera is represented by the number 2 and a darker colour. This representation is the area that is captured in the photograph. The field of view of the context camera is marked by the number 1 and represented in a light colour. This is the area where it is determined if there is a passing vessel. This translates in little space and time to detect the vessel in time to take the photograph when the vessel is passing in the conjoint field of view. The direction in which the vessel is moving is also a parameter to obtain. There were two alternatives considered: object tracking or register if the movement was detected in the left or right side of the field of view.

The test consisted at first in videos gathered from the internet, in a second phase blender animation. In this animation obstacles and weather changes could be manipulated (e.g. undulation or fog) as well as camera angles providing a

good and extent test case scenario. Lastly the system was tested in field. Below are described the most successful approaches:

- **Kalman filter**

Kalman filters are one of the most robust and fault-tolerance solutions in a wide range of applications (Moeslund, T. B., Hilton, A., Kruger, V., & Sigal, L. (2011). In image processing, it is very commonly used to detect and track changes in a video feed, i.e., moving objects. The kalman filter is always adjusting to the current conditions, which make it very sensible to small changes in almost static environments and less sensitive to small changes when the environment is dynamic. This adaptability makes the kalman filter an ideal method for tracking. Using a simple thresholding of the areas with detected movement work can be used most of the times to eliminate high sensibility detections. When a moving vessel approaches, a big area pops up, triggering the system to take a photograph, and using the center of the detected blob, it can be used to track the vessel location. From all the algorithms tested, using a kalman filter is one of the approaches with higher performance and reduced complexity, using libraries such as OPENCV. The downside of this method is the need to keep a series of mask images, which using a C++ DLL is a computer intensive task, since a DLL call cannot store objects, forcing the objects to be constantly passed from the calling program to the DLL. This minimizes its performance critically to the point where some of the vessels were lost, due to the limited size of the field of view of the context camera.

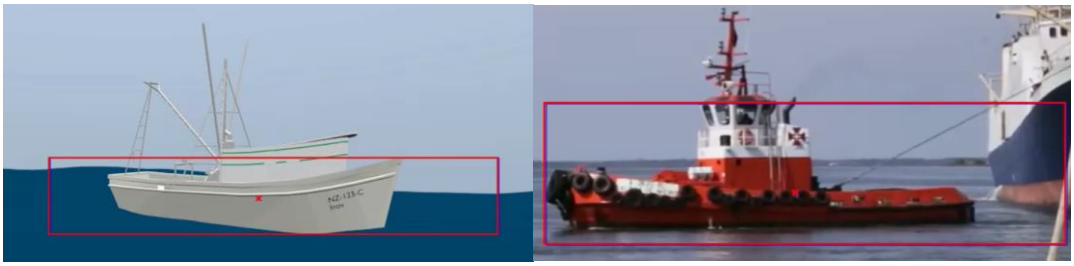


Figure 16 – Example of the kalman filter aplied to a simulation made in blender (left) and a video (right).

- **MOG Background subtraction**

This approach is similar to the kalman filter technique, mainly due to the image subtraction eliminating most of the background of the image. There are several image subtraction techniques, including different version of the MOG background subtraction, depending on the desired sensibility. In this approach, a mask image is created, able to store the similarities of the provided images. This mask works similar to a stack or queue, where a parameter defines the maximum number of images it holds. The greater the number of images stored the more accurate it becomes. And for about 500 images stored in the mask, the algorithm's performance is accurate enough to detect big moving objects, i.e., passing vessels. Since this solution is very similar to the kalman filter, both the

performance and downsides are the same. The need to keep the masked image, and constantly pass it to the managed code limits its efficiency, making it lose some of passing vessels, due to the processing delay.

- **Pixel-based detection**

Tweaking the image using certain image processing functions is a quick and valid way to segment a moving object. While other methods tend to adapt to the circumstances, pixel-based detection must be done carefully because it is easier to adapt a method for very specific cases. Several approaches were tried, all based on pixel-based principles, most of which didn't achieved the desired performance. Unlike with the kalman filter and the background subtraction, these approaches took in consideration the method is used by a DLL, which means no objects can be stored in memory, and must be as fast as possible. Some of the approaches track the vessel, while other rely on the origin of the trigger (rather if it was triggered on the left or right side of the field of view) or comparing the coordinates of the biggest contour. Below are presented some of the most relevant implemented/tested approaches:

First approach:

Input: two successive frames

Output: Ratio between number of white pixels and total pixels

1. Image subtraction;
2. Grayscale conversion;
3. Binary thresholding;
4. Pixel counting;
5. System trigger.

Performance: It was given an adjustable threshold parameter, still no perfect value was found. That is if the value was too low, it captured a lot of false positives (e.g. waves), and for higher values some of the smaller fishing boats were undetected. There was no way to tell which direction the vessel moved besides the origin of the trigger (left or right-side).

Second approach:

Input: two successive frames

Output: Ratio between number of white pixels and total pixels; coordinates of biggest contour;

1. Grayscale conversion;
2. Image subtraction;
3. Intensity thresholding;
4. Gaussian blur;
5. Binary thresholding;

6. Contour detection;
7. Store coordinates of the biggest contour;
8. Pixel counting;
9. System trigger.

Performance: It was able to track the vessel, but a perfect threshold value was still difficult to find, and as a result, some vessels weren't captured and there was a notable number of false positives.

Figure 17 – Output displayed after running the algorithm on a video. In green are represented the centroids of the object for each frame processed, and yellow, to join the centroids, the estimated path covered by the object.

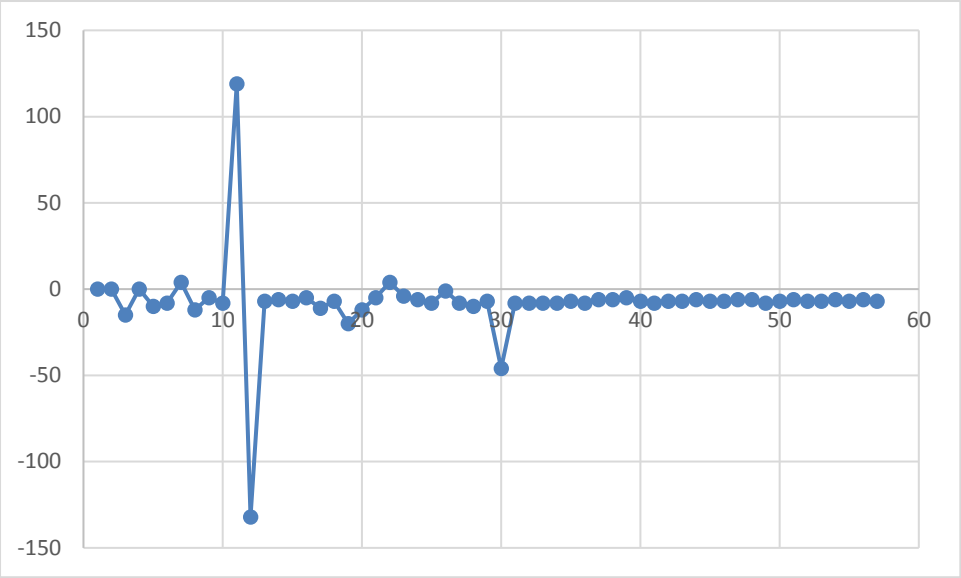


Figure 18 – Graphic representation of the location of the centroids of the previous image. In the x-axis are represented the processed frames and in the y-axis the difference between the positions in the XX axis of the centroids, from one frame to the next.

Third approach:

Input: two successive frames

Output: Ratio between number of white pixels and total pixels; coordinates of biggest contour;

1. Gaussian blur;
2. Image subtraction (in RGB);
3. Grayscale conversion;
4. Contour detection;
5. Store coordinates of the biggest contour;
    a. Optionally, comparing the shape of the contour with the previous contour - improves accuracy when the contour that defines the object is stable, but worsens if there are differences in the contour
6. Pixel counting;
7. System trigger.

Performance: It was able to track the vessel and had a better overall performance. But attending to the limited space of the field of view of the cameras this approach couldn't be implemented in the final version of the system.

Figure 19 – Output presented after the execution of the 3rd approach in the same video.

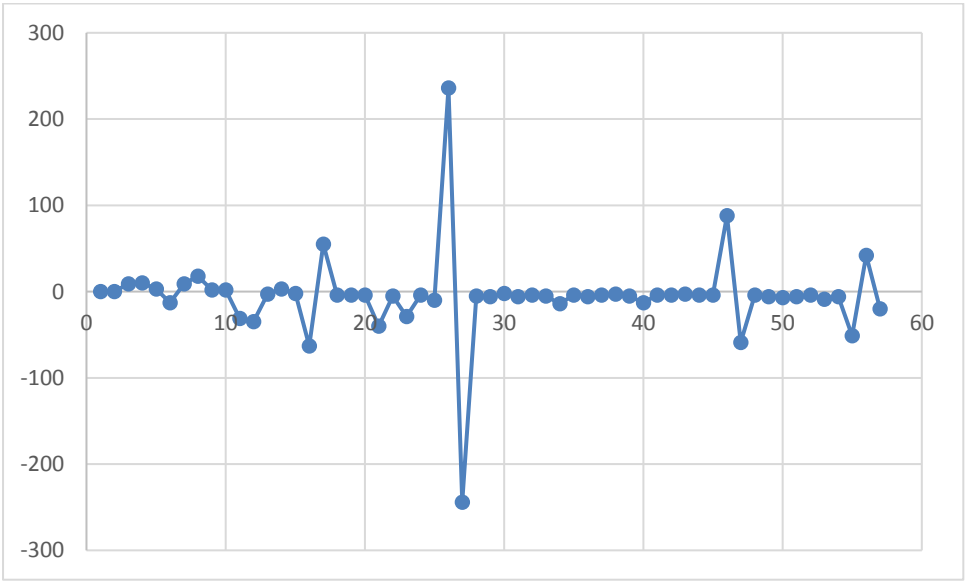


Figure 20 – Graphical representation of the differences of the coordinates of the centroids calculated using the 3rd approach. A notable improvement can be noted comparing to previous approaches.

Forth approach:

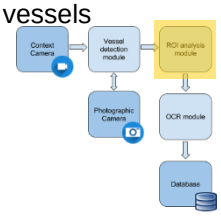
Input: one frame, a background image and 3 regions of interest, 2 denoting the zones from which the vessels can appear and one to detect the color of the water;

Output: Ratio between number of white pixels and total pixels; coordinates of biggest contour;

1. Resize the image to remove noise (waves, birds, etc.) and processing time;
2. Background subtraction;
3. Gaussian blur;
4. Convert the image to HSV colorspace (hue, saturarion, value);
5. Detect the hue color of the region containing only water;
6. Subtract blueish colours using the hue;
7. Grayscale conversion;
8. Binary thresholding;
9. Exclude regions outside regions of interest;
10. Erosion and dilation until obtaining the skeleton of the contours;
11. Contour detection;
12. Contour filtration based on aspect ratio and area;
13. Store coordinates of the biggest contour;
14. Pixel counting;
15. System trigger.

Performance: Attending all previous results, this one represents a great improvement, since it reduces a great number of false positives, while capturing most passing vessels (85%). Depending on the first coordinates, the direction of the vessel is calculated. Subtracting blue colours, using the hue of a known area which contains only water substantially improved the algorithm.

In conclusion, there is a trade-off between the resistance to noise and sensibility, which causes most solutions to either be very sensitive to weather changes or birds, or to miss passing vessels. Since field tests are very limited it is very difficult to predict certain non-tested scenarios such as fog. From all the tests, it was concluded the kalman filter clearly had the advantage and worked almost flawlessly. On the other hand, since we're working with a DLL and it is not possible to maintain objects in memory. The algorithm substracts multiple images through time and since it is not possible to store all this operation in a single object, using opencv's Kaman filter implementation proved to be impossible to implement in the system. As such, the second-best approach was chosen, the approach 4, which used the regions of interest and pixel-based detection to trigger the system. Its performance is currently as 85%.



c. *ROI Analysis Module*

To evaluate this module, a considerable database was gathered, consisting in 589 images of vessels and more than 2500 negative images. Keep in mind, all these images had to be manually processed to identify the region of interest for the training algorithm. Also, a lot of the images had to be filtered out of the dataset, because they didn't fit for the purpose, e.g. they were not Portuguese professional fishing vessels or the image quality was not sufficient for a clear reading of any of the ship's registrations. The following approaches have been attempted for the detection of regions of interest:

- **Cascade Object Detectors:**

As both sets of images (positive and negative) were being collected, several classifiers with different hyperparameters were being trained. There are 4 main hyperparameters that had to be adjusted:

- Number of cascade stages – The number of stages to train. Increasing this variable allows for a more precise classifier, but increases the training time and the number of images needed;
- False alarm rate – Corresponding to the maximum acceptable number of false positives per stage, varying from]0,1]. Decreasing the value of false alarms increases complexity by stage and increases training and detection time, with the advantage of reducing the number of false detections;
- True positive rate – Corresponding to the minimum number acceptable to finish a stage, varying between]0,1]. Increasing this rate increases the performance of the classifier, but also increases the complexity and times of training and execution of the classifier;
- Negative samples factor – Determines how many negative samples are used in each stage, and their default value is 2.

Below is an example of the classifier training process, starting with a negative image, a positive image, and the concatenation of both, generating a new training sample. In this way, it is possible to increase the number of positive images to train the algorithm.

Figure 21 – Training process using cascading classifiers - A negative image (top left) is concatenated with a positive image rotated, scaled and translated (top right) resulting in one or more sample (s) that can be used to train the algorithm.

- **Haar Cascade classifiers**

Of the various tests that were being done, the Haar classifiers proved to be very slow to train, taking as much as four times longer than the other two options. The result obtained was not very relevant because it contained too many false positives to be considered useful. The run time after training was already acceptable, but it was still less efficient than the other sorting types. Figure 22 reveals some results obtained after the use of the algorithm. The same images are used for comparison between the various classifiers.

Figure 22 – The result of the application of a Haar cascade classifier, trained with False Alarm Rate of 0.5 and 20 stages completed.

- **LBP cascade classfiers**

Similar to the Haar classifiers, LBP classifiers have many false positives and are not particularly useful for the study. The result of this type of classifiers is the worst among the three types of classification, failing to

detect plates and/or having such a high number of false positives that little help in image processing. Figure 23 contextualizes some of the results obtained.

Figure 23 – Result of the application of an LBP classifier, trained with False Alarm Rate of 0.5 and 20 stages completed.

- **HOG cascade classfiers**

Several HOG classifiers were trained from about 120 positive images up to 420, and from 1000 negative images up to 2150. From several tests, it was verified that the true positive rate should always be kept at 1, and the algorithm should Always identify at least one of the regions of interest per image for each stage. Therefore, the performance of the algorithm would be essentially dependent on the other two variables. The false alarm rate should be kept as low as possible by the number of images. The number of stages was defined as 40, never reaching this value. In this way, it was guaranteed that the algorithm would go through as many stages as possible. The training time was fast compared to the other classifiers and the execution time was only slightly different. Figure 24 summarizes some of the results obtained in this type of classification.

Figure 24 – Result of the application of a HOG classifier, trained with False Alarm Rate of 0.5 and 20 stages completed.

- Support Vector Machine (SVM) with Bag of Features

This approach requires the use of at least two categories of objects. The idea behind this approach is essentially to distinguish images and group them into categories. This is a different problem from the project at hand since we want to determine if there is a boat and regions of interest in the image, and not choose which category of the image.

- Part-based classifier

This approach uses object-based classification to detect parts that make up a vessel. Initially the algorithm would search for the cabin, the prow and the stern. Several classifiers were trained, such as those shown in the object classification section and the HOG classifiers proved once again to have a superior performance. This classifier was trained to detect the prow of the vessel. However, its performance wasn't as successful as training an object classifier for vessel plates, failing for more than 50% of the vessels. As such, this approach was rejected. Below some images of the result of the classifier.

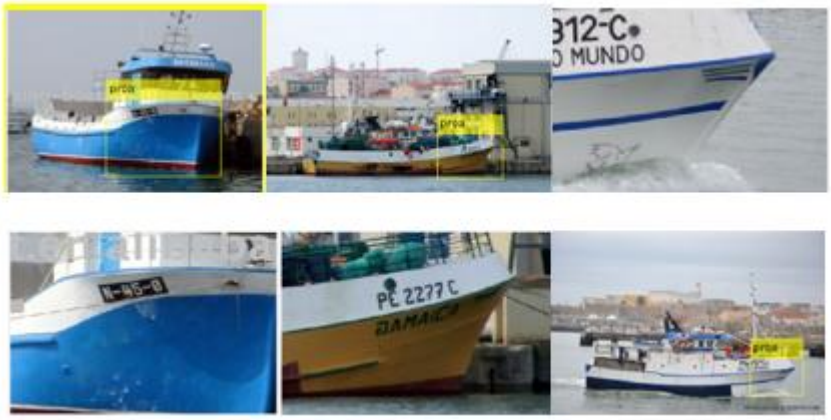


Figure 25 – Result of the application of a HOG classifier, trained to detect the prow of the vessel.

- Pixel-based classifier

The first step of this approach is to convert the input image to a grayscale image, as shown in the image below, and identifying the zones with the more relevant geometric properties. To do so, we extract the MSER (Most Stable Extremal Regions) features present in the images below.

Figure 26 – Input image (left) and the corresponding grayscale convertion (right)

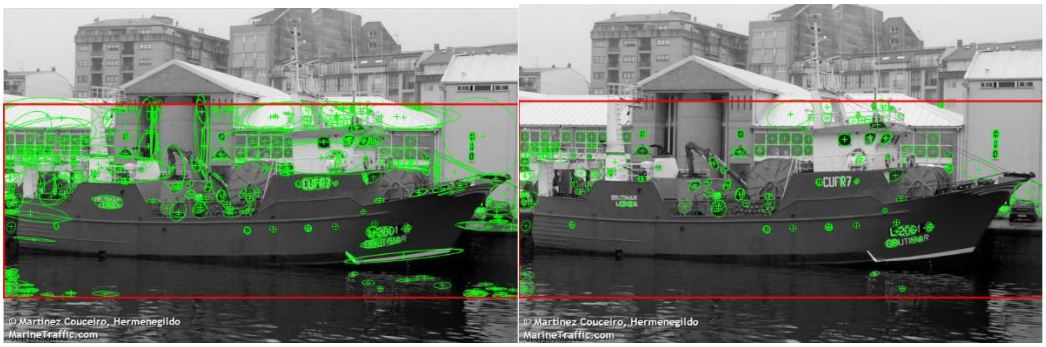


Figure 27 – Representation of the MSER features pre-filtering (left) and post-filtering (right)

The next step is to filter all the non-relevant features in the image according to their geometric properties. A zone where the boat is located was defined as a red rectangle in the image and represents the contribute from the previous module, which can be used to reduce some of the non-relevant parts of the image. Two main sets of rules were applied to the MSER features in two separate attempts, one stricter in order to reduce false positives and another less strict in order to detect the most amount of text possible: Exclude all MSER features with eccentricity greater than 0.9, solidity lesser then 0.6 or Euler Number lesser than -3, since they don't fit the typical character features, according to Matlab documentation regarding detection of text in natural images. This proved to be a stricter set of rules and exclude some of the text and vary few false positives. - Exclude all features with eccentricity greater than 1 or lesser than 0.2, solidity greater than 1 and lesser than 0.4, Euler Number greater than 1 or smaller than -20 and with aspect ratio of the bounding box greater than 5 or lesser than 0.5.

These values were obtained by creating a bounding box surrounding all the desired features and extracting the maximum and minimum values of all the features detected in those regions. This set of rules on the other hand tend to include most of the relevant regions as well as some false positives. Next, the bounding box surrounding each region is expanded and the neighboring regions overlapping are merged, generating bigger bounding boxes. The goal in doing

this is to find words and phrases, carrying each word meaning and considering the order of the characters in the word is as important as finding the character itself. The figure below represents, in yellow color, the merged bounding boxes.

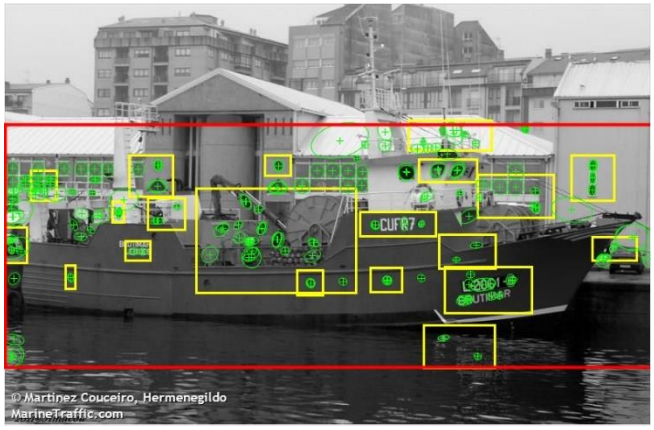


Figure 28 – Representation of the MSER features and the merged neighbouring bounding boxes

Figure 29 – Representation of the 100 strongest corners (Harris Features) in the image

After that, a Harris feature detector was applied in order to find the 100 strongest corners, present in the image, such as represent in Figure 29, followed by eliminating all the previous bounding boxes that didn't contain a certain number of corners. Two approaches were applied (shown in the images below): at least one corner present in the region is necessary (left) and at least 4 corners were necessary (right). In this conditions the solution could always present results in under 3 seconds.

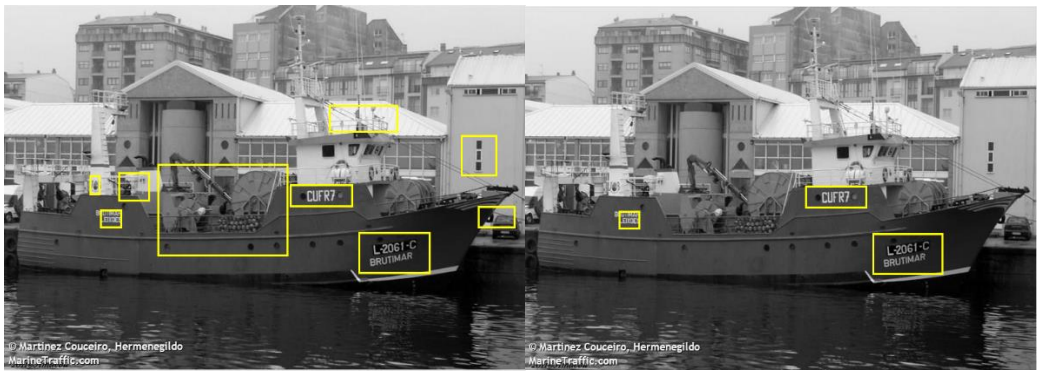


Figure 30 – Final output with regions containing 1 or more corners (left) and 4 or more corners (right)

Other tests were also performed, such as removing the area of interest that would be provided by the previous module (red rectangle), shown in Figure 31 on the left and filter by the percentage of corners contained per area, instead of counting (Figure 31 on the right). To do previous calculation, the number of corners in a region is divided by the area of each region and if this value was lesser then 0.0003 the regions was excluded. In that test, the size of the rectangle was also increased, only eliminating the photographer's name in the image.

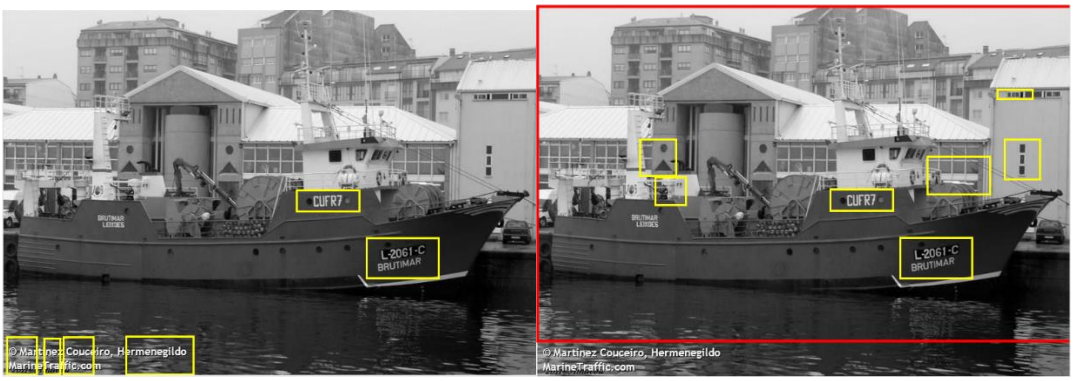
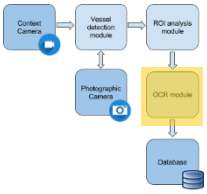


Figure 31 – Final outputs without any region of interest to aid the solution (left) and with the rate of corners per area (right)

- Convolutional neural network (CNN)

It was also developed a deep neural network using MATLAB's toolkit but with the current size of the dataset, the results were much poorer than it would be expected. The network had a success rate around 60%. A bigger dataset had to be used to achieve the same or greater success rate of previous approaches.



d. *OCR Module*

For the OCR component, 2 tools were used:

- the MATLAB OCR tool - used when the algorithm was in the testing phase and only implemented in MATLAB;
- Tesseract-OCR - an open-source OCR engine sponsored by Google and currently known as one of the best free according to OpenCV communities (Tesseract-OCR, 2017).

Free alternatives were also compared on the Internet, but the performance was too low since the purpose was to extract characters in scans of photocopies. Currently available solutions are all paid and proprietary with little to none information available, most of which are designed for non-skewed text. Due to the budget for the project, proprietary solutions are not considered. From open-source to proprietary, the Tesseract is in the basis of most of the tools and as such most of the changes are made at the image pre-processing level. MATLAB OCR tool and Microsoft's OneNote were the only proprietary tools tested, but since no licenses were made, these tools were solely used for testing purposes. After several attempts to use these tools in several scenarios (original images, manually processed images and images pre-processed by developed algorithms) it was concluded only very small pieces of information could be retrieved and a lot of incorrect information was incorrect. The tesseract library used was not equipped with any pre-processing tools. All the approaches made in this section are pre-process algorithms to improve the library's performance.

Alternatively, it would still be possible to develop a new OCR tool, but such an approach would be a difficult and time-consuming task, so it would be an approach to add to future work but rejected taking into account the time available for the project.

Recognizing the characters on these images is a complex process due to the lack of standards. The process on vehicles plates is easier due to the existence of standards - the registrations are printed with a low-reflective foreground and a high-reflective background. Most of the problems arise from letters of different size which most of them are skewed and/or distorted, the colours used and a curved shape of vessel prow.

Given the lack of standards, the lack of colour contrast and the available (non-proprietary) OCR tools, any of the proposed solutions have a low performance in natural and unprocessed images, so the use of pre-processing is essential. Using more rigid standards such as those of the car plates would make it easier to detect the plate numbers. The library used can be parameterized to a certain language or to set regions of interest, but since the images are already cropped and vessel plates don't fit in a specific language, most of this module consists in pre-

processing to allow better results when using the library. Several approaches were tried:

- Approach 1:

The first step is to convert the image to grayscale. This is done mainly because the legislation indicates the character's colours must be contrasting to the surrounding colour. With a grayscale conversion this distinction is easier. The second step is to apply the OCR tool to the image, save the results to a list, and rerun while applying rotation on each iteration. This cycle would be repeated within and a predefined set of angles. The angle would be chosen taking into account the extracted characters, which would be attributed a score based on the number of valid characters (avoid punctuation or special characters) and compliance with the rules of registration.

This approach is far from being the most adequate, since it's a typical brute force approach. Most characters were not recognized and the performance improved a little when the results from all the rotations were combined, since some characters had a different rotation than others.

- Approach 2:

This approach starts by converting the image to grayscale. The MSER features are extracted and filtered using fixed sets of rules. The bounding box of each of the remaining feature is calculated and expanded. Only intercepting boxes are kept and grouped together. Using linear regression and the centres of the bounding boxes the rotation angle is calculated and the image is transformed. The OCR tool is executed. Alternatively, the image was also binarized to minimize the noise in the image.

The performance of this approach is also limited: there were some cases where the text was simply not found and there were cases where almost all the registration was detected as text. As for the extracted text, it contains noise and its performance was not the desired. Below are some examples of this approach.

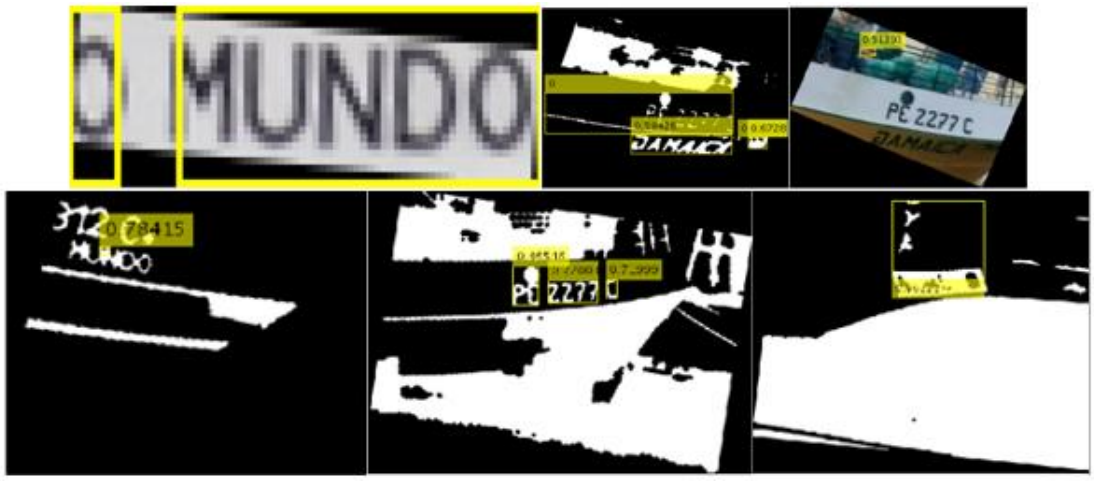


Figure 32 – Result of the application of the OCR tool in several of the tested scenarios. The confidence factor is shown in a yellow text box.

- Approach 3:

In this approach, the image is converted to grayscale and a copy of the image is made. A bitwise-not operation is applied to the copy of the image, inverting its colours. Since the OCR library assumes the text to be darker than the background, having an inverted grayscale image guarantees that text with a brighter colour is detected.

The images are then binarized. Several binarization algorithms were implemented and tested: adaptive, Otsu's, Niblack's, Sauvola's, Wolf and Jolion 's and thresholding using histogram filters. From the test made, Otsu's thresholding algorithm proved to be able to best segment the characters from the background without losing much information. These results are very difficult to quantify due to the diversity of results presented. Initially it was taken in consideration the amount of background area removed and the amount of foreground area not removed but each of the tests revealed very different results depending on the input image.

After the binarization, the MSER features are extracted and a fixed set of rules filters most of the regions, not corresponding to regions with possibility of containing text. An adaptive threshold approach was also included in some of the approaches, to compensate some of the previous binarization algorithms with lower sensibility. Contours of each region are calculated and again filtered by a fixed set of rules. An auxiliary software was created to calculate the maximum and minimum of MSER features values and sizes of contours. The software works by calculating all the MSER features that fit inside a bounding box created by the user and filters out all the features that do not fit to obtain most of the characters of a given registration. The image set used in the software consisted of 200 images. The sets of rules used are created according to the legislation and rules for vessel registrations as well as the values calculated by the auxiliary software.

Figure 33 – Output of the algorithm. In this case, the character segmentation was disabled and only the rotation and deskew operations were enabled.

- Approach 4:

This approach begins by converting the image to grayscale. To find the skeleton of the image erosion and dilation operations are applied to the image and both results are combined, adding both masks. These operations are applied

to the extent that the result of the next iteration is a completely unique mask, i.e. consisting solely of zeros. The last iteration before the mask becomes only composed of zeros is the resulting mask. This gives you the strongest traces of the image.

Next, the contours of the skeleton are calculated and the bounding boxes of the contours determined. The MSER features of the image are calculated in grayscale and filtered according to a set of rules: height must be greater than width, aspect ratio and dimensions must be within a limit calculated by the mean and standard deviation of the width and height.

Figure 34 – Input image (left) and image after the segmentation (right)

Both processes return masks, which are then multiplied in order to find all intersection points and generate a new mask. The mask is dilated to ensure that no characters are cut off by any effect such as shadows or other noise. The mask is applied to the original image. Using the average of the centers of mass of the bounding boxes is calculated the deskew and run the OCR engine. This approach has proven to be the most effective both to segment and to detect the deskew, however the OCR result remains very imprecise. For the images presented below, the top row of the text was correctly identified as “PV-276-C” but the in the bottom row not a single character was correctly identified.

Figure 35 – Result after operation of rotation (left) and after deskew (right) using the forth approach

C. Boarded System

Regarding the boarded system, a camera is linked to the VMS system, whose functioning is outside the scope of this thesis. The VMS system (MONICAP) is placed in one of the highest points of the ship, allowing the system to monitor the aft/stern platform. Figure 36 puts it under perspective. All the data is transmitted to the authorities when in range, but all the image processing is processed in the VMS running the solution. Therefore, the implementing method must be as light as possible. The system is composed of only one module responsible for receiving the data from the video camera, identify and count the fishing gear and track the objects. The algorithm must be able to notice drops in the number of fishing gear in the vessel, signalling the start of a fishing activity. The inverse is applied to signal the end of the fishing activity.

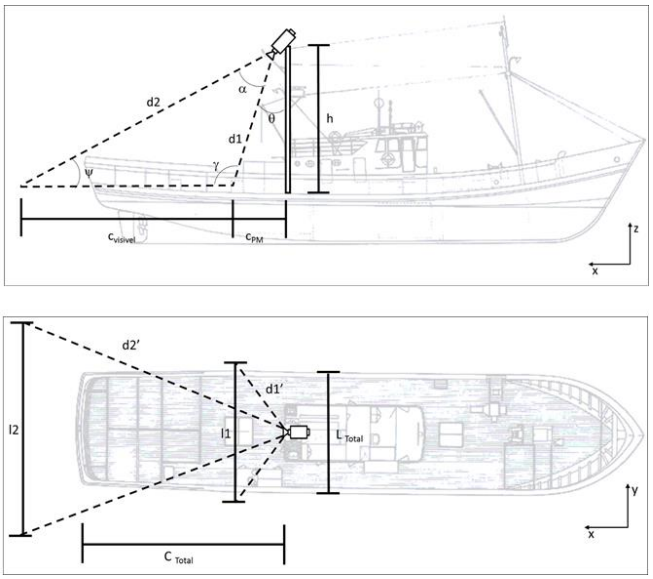


Figure 36 - Schematics of a regular fishing vessel and camera position

The real-life result of the implementation is represented in Figure 37. Since the camera's position and orientation are a fixed variable and since the current hardware needs configuration, defining regions of interest eases the processing of the image, since only a portion of the image needs to be analysed. In the real-life test, only the left side of the camera's field of view.

Figure 37 – Example of an acquisition of the camera

As previously mentioned, this system is composed of a DLL which is called by the hardware's operating system at the configured times, and has to work with only one image. This is a huge limitation since there is no control of the time the image is taken nor access to an history of the previous positions of the vessel's fishing gear. On the other hand, since the gear is generally placed and manipulated in similar fashion, we can observe a limited number of colours: the colour of the sky and water, the colour of certain items/clothes, the colour of the vessel's "walls", the colour of the floor and the colour of the gear. Converting the image to another colorspace and applying a segmentation method such as K-means can separate the image in a useful way. With five clusters, we can segment the image by the colours previously enumerated. This way we get five different images, one for each of the segmented colours. The next step is to filter out the non-relevant images. We are looking for a big area extending to at least half of the image's region of interest. Plus, since the gear is a net, it is expected a big number of holes in the segment image, corresponding to the net's pattern. Summarizing, the segmented image has a big area, which eliminates the 3 clusters with fewer areas, and with the most "holes". Calculating the Euler Number of the image provides a negative number corresponding to the number of holes detected, that is, the smaller the number the bigger the number of holes.

However, the gear may be missing and with the current method there is always a chosen cluster. To avoid false positives, it is necessary to apply another filter: a threshold to the number of holes detected. After analysing multiple images, a calculated value was determined and was later rounded to -900. That means, the segmented image must have at least 900 holes to be considered a fishing net, otherwise it is considered a false positive. The figures below show the input image, the region of interest detected and the segmentation of the fishing gear (net).

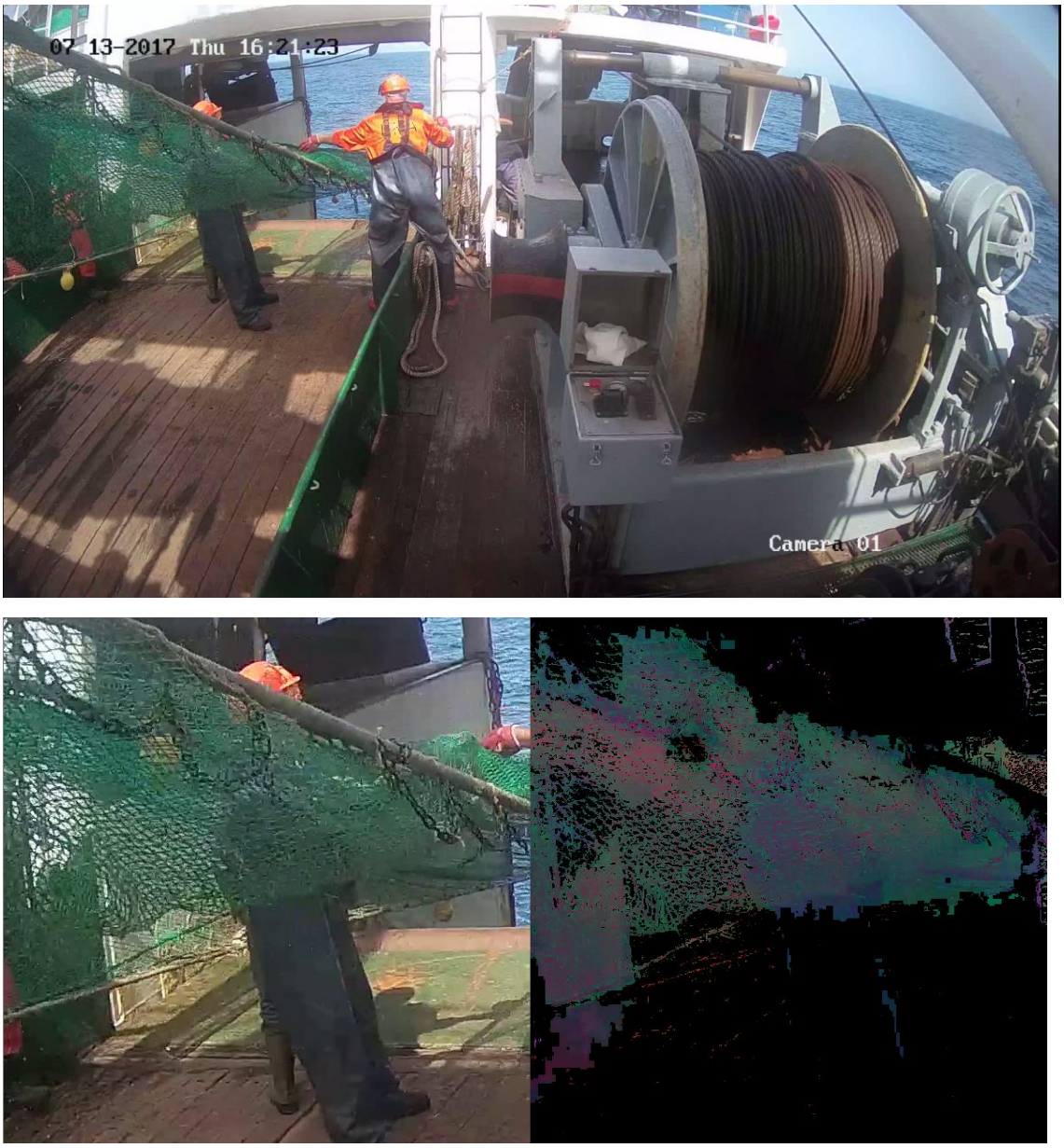


Figure 38 – Application of the algorithm in the input image (above) and corresponding results with segmentation (right) and without segmentation(left)

IV. COMMERCIAL SYSTEM

Both the non-boarded system and the boarded system are prototypes that have been tested and installed in both simulation and field environments and consist of a larger part of a commercial and research project. The SeaITall project aims to provide the existing control center (Seasight) with new improved vessel control features. There are several components that constitute the project, but only two components are studied in this thesis: the non-boarded component is destined to the monitoring of inputs and outputs of the port, and the boarded component is destined to the monitoring of fishing activities.

A. Non-Boarded System

In the final implementation of the system it is necessary to take into account some aspects that had not previously been considered:

- The machine where the algorithm is created and tested and the machine (s) where the algorithm is applied have different performances, leading to the need to test if the algorithm is light enough so that there are no performance problems in machines with reduced performance;
- Robustness, both due to hardware and software failures, but also through careful management of memory and disk space;
- Time and ease of implementation / conversion of the MATLAB algorithm to C ++, since there are highly sophisticated and / or optimized methods whose C ++ replication could take too long;
- Communication with the outside, that is, regular communication with the control center is necessary through a web service.

As previously mentioned, the non-boarded system consists of an executable written in C #, which integrates and interconnects the various components, playing an essential role for the general operation of the project: communication with the C ++ DLL and evoking its image processing methods; Communication with hardware equipment (both cameras); Management of local resources, including disk space and memory; Communication with the authorities and the control center for the exchange of information. Figure 39 contextualizes how the various components are linked.

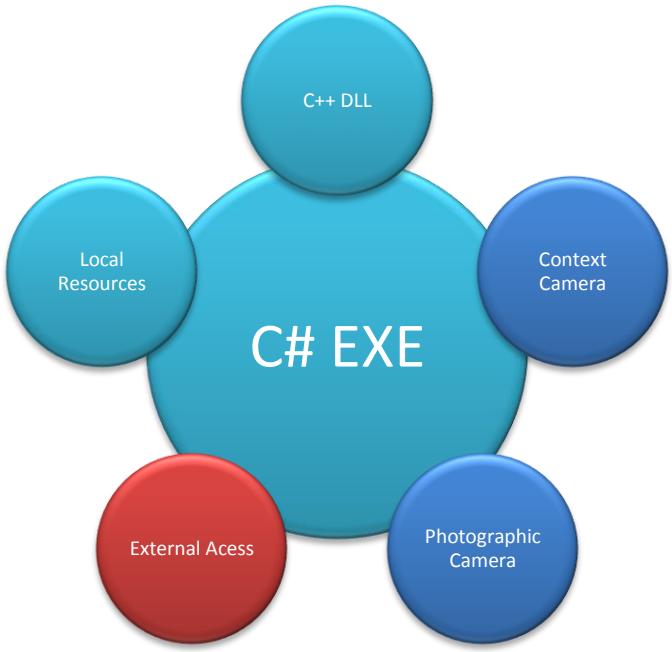


Figure 39 – General outline of the non-boarded system. In a lighter blue color represents the main software components that should intercommunicate with each other, in dark blue the hardware components with which the software interacts and in red the interface (web service) that is available to the outside.

- **External Access**

External access is via web service. The communication is made using the SOAP protocol, rather than REST, due essentially to the large number of existing standards (formal contracts), ensuring the security and reliability of communication. The same advantages are also guaranteed by the fact that processing and calls are done asynchronously. The business solution relies on the control center as a client and the solution developed as the server. In this type of architecture there is pooling, that is, it is necessary for the client to question the server if there are new events, which can produce overhead if there are several clients connected to communicate constantly. On the other hand, it is a more robust solution, which allows the existence of several clients and simplifies the entire architecture. Table 4 contains the methods used in the communication between the system and the control center.

Method	Function
GetVersion	Obtain software version for compliance verification purposes
GetSystemStructure	Obtain the structure and capabilities of the system hardware components (ID, location, orientation and capacity/ performance)
GetRealTimeImage	Get real-time image of system hardware
GetEventsByDate	Get detailed list of detection events given a start and end timestamp
GetEventImage	Get the image of a given event

Table 4 – Name of the webservice methods and their functions. The table is organized so that the methods at the top obtain data that can be used as arguments to methods below.

- **Hardware/Cameras**

After evaluating the advantages and disadvantages, and as previously indicated in the section of the methods and system developed, the location chosen for the installation of the cameras was next to the center of the port and further away from the sea. This location is more advantageous than the other location and allows the choice of specialized hardware for the present conditions. The performance of the cameras is limited to the following rules: 1) the speed of the vessel must not exceed 5 knots; 2) the distance from the vessel to the cameras must be within 40 and 95 meters, in optimal weather conditions (e.g. no fog or rain); 3) characters of the plate must have a height between 30 and 100 cm. Keep in mind that despite the legislation about the characters in the registration, most vessel registrations do not follow all the rules and standards. The same is about the vessel speed limits in the ports, which is why some of the results were excluded in final tests, since they represent illegal situations. Below is presented a list of properties of both cameras.

Context Camera	
Light sensitivity	0,017 lx
Resolution	720p (HD)
FPS	60 frames/second

Table 5 – Context camera specifications

Photographic Camera	
Sensor	CMOS
Resolution	10.55 Mp
FPS	6.1 frames/second (max)

Table 6 – Photographic camera sepecifications

- **Local Resources**

The maintenance of local resources is done by the application developed in C#. There are essentially two resources to manage: memory and disk space. Since the component developed in C++ is a DLL, all objects created during execution are erased from memory after each call, and no further maintenance is required. The component developed in C # is optimized to eliminate many of the created objects from memory, through the use of the garbage collector of the .NET framework. The disc is occupied with the videos and photos of each detection event. This information is then sent to the control center. There are a few options to consider when cleaning files: Files can be deleted after they have been sent to the control center; Or may be maintained for a given time period (e.g., 30 days). Both approaches have been implemented, since both have advantages: the first approach allows a better memory management, but does not result in the presence of several control centers or in the event of communication failures; The second approach allows information to be maintained even in the event of communication failures, but there is no controlled memory management. The solution implemented consists of the second approach, maintaining the files over a period of time, but monitoring disk space and switching to the first approach, delete sent files, after exceeding a tax threshold.

- **C++ DLL**

A DLL containing all the image processing methods written in C++ and using the OpenCV DLL.

- **C# Executable**

The executable that manages all the remaining components.

B. Boarded System

The boarded system implementation was delayed due to the MONICAP (VMS system) problems. All the algorithms and final systems are already compiled in a C++ DLL to be run by the VMS system and are waiting for the final release of the system. Details of the integration cannot be displayed and are not in the context

of this thesis. The DLL receives images as parameters and outputs how many fishing gears it detects and confidence factors for each.

V. RESULTS

This section evaluates the solutions and assumptions previously elaborated. Recalling the questions made in the introductory section, the goal here is to discuss which of them fit the requirements (robustness, performance, etc.) for the elaboration of each of the systems, noting they may be disparate.

A. Non-Boarded System

The non-boarded system, on the one hand, requires a fast algorithm for the vessel detection, allowing the capture of multiple boats in a queue to enter or exit the harbour. The system also needs to be robust against other moving objects.

a. ***Vessel Detection Module***

Attending the previous section, to keep a real-time performance and considering the limitations of the release system, the implementation had to be a pixel-based approach. Taking into account all the tests done, either in laboratory as well as on the field, the results suggest the forth approach is the most appropriate.

This approach took advantage of previously marked regions of interest: one marking an area of the water which had no vessels (the shallow zone of the passage); two regions where the vessels could be detected, one for entering the harbour and other for exiting the harbour. The two regions for the vessel detection have around 25% of the total width of the field of view, corresponding to the area in which the vessels have to be detected, while the other 50% are left for the region of the photography, also taking into account the time it takes to activate the second camera. In other words, if the vessel is not detected in those two regions, then when the detection is made, the photo will not fully contain the vessel.

This final release of this module was implemented on June 2017 on the Nazaré's harbour and has been continuously running. Within these 3 months a lot of data has been gathered, corresponding to more than 2500 detections, and after an analysis it can be concluded this approach had a great performance in field. Around 15% of the captures were triggered by shadows, waves, birds or lightning changes, corresponding to the false positives, and the remaining 85% being vessels of any type. Unfortunately, most vessels were not fishing vessels, which means there is still much room for improvement. On the hand, since the goal of this module was the detection of vessels and no specifically fishing vessels, this module proved to be a success. There is a need to add new functionality on this module: verifying the vessel type. This functionality is now being worked on and will remain as future work. The triggered window indicates

the direction of the vessel making it 100% accurate for the vessel captures. This solution seems satisfactory for the short window of time available for the processing. Below is a sample of the images captured by the context camera, and a representation of where the regions of interest are located.

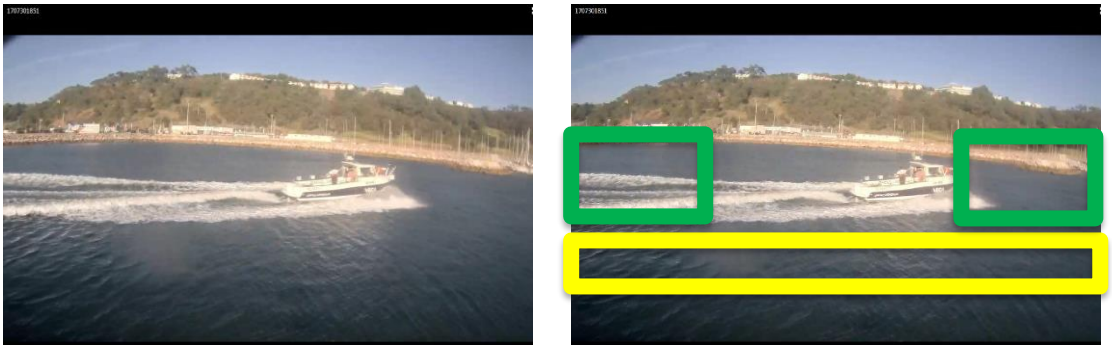


Figure 40 – Sample of the context camera image (left) and representation of the regions of interest (right). Marked with green are the detection regions where movement will be detected, and marked in yellow is the region containing only water for hue comparison

After the detection of the vessel, the system triggers the detection camera to take a photograph and the ROI Analysis Module is executed with the same photograph as parameter. The failure to detect a vessel implies the failure of the whole system since no inputs are passed to the other modules.

b. *ROI Analysis Module*

This module receives as input a high-quality photograph, triggered by the previous module. The main difficulties were external to the module and were related with the hardware.

There were difficulties in timing the photograph, mainly due to the irregular delay in the activation time of the camera and the added delay to allow the whole vessel to enter the camera's field of view which sometimes took too long to activate. Sometimes, it also malfunctioned outputting incorrect images, such as presented below. Note the bottom half of the image is updated while the upper part of the image is from a previous image left in the buffer.

Figure 41 – Example of a malfunction of the detection camera

Other difficulties were faced such as finding the appropriate acquisition time to avoid having an image too dark or too light. A lot of time was spent on configuring and test the camera settings such as aperture, shutter speed and ISO. As such, there are less accountable results for this module.

On July 2017, this module was implemented in the system placed at Nazaré's harbour. This module works together with the vessel detection module since then.

Despite the laboratory results, applying the classifier is a reliable solution obtaining almost all the visible text in the photos tested. The missed texts were illegible because of light reflection on brighter surfaces. All illegible text was correctly ignored. The performance in field was around 95% for more than 1500 detections. This performance values only considered detections containing vessels, since false positives from the previous module are errors that should not influence the results of this module's performance.

Below is shown an example, in which there were several activations: 3 outside the vessel but corresponding to text in the buildings on the background; 3 corresponding to actual text in the vessel and corresponding to correct activations; and 1 incorrect activation near the bottom of the vessel.

Figure 42 – Example of a successful execution of the algorithm

Due to the high-quality image, some small advertisements on the background building were captured. This condition was not predicted before the implementation on the harbour and similarly to the vessel detection module, false positives deriving from the advertisements were not considered. This leads to a need to filter out the vessel before processing it with the classifier. This feature can later be added and will be considered as future work, since a new classifier may be needed. As for a temporary solution, the area of the region is thresholded using a region of interest which excludes the buildings in order to avoid such background noise.

After the detection of the regions of interest, each region is cropped and analysed by the OCR module. The failure to detect a region will imply the failure to detect a text.

c. *OCR Module*

This module receives as input several images corresponding to each of the areas cropped by the previous module. This module considers each input to be an image containing text. Any image not containing text is considered a failure of the previous module and as such this kind of errors were not considered in this module's performance.

Due to its performance, the final version of this module is yet to be implemented in the system located at Nazaré's harbour. A provisory version was implemented in the harbour to obtain some field results for belter understanding and for future improvement of the module.

This module is composed of 2 parts: the pre-processing of the image and the text extraction. The first part is responsible for segment the text in the cropped image, calculating the correct rotation and deskew angles and applying these transformations to the image. The second part is made essentially by calling a OCR library (tesseract) and extracting the text present in the image.

The performance of the pre-processing of this module is good. For a dataset composed of 200 images, 82% of the characters were correctly segmented. Most failures occurred due to: shadows which altered the colors of the characters; rust or deformities on the hull of the vessel; excessive brightness; text size variations in the same word. The rotation angle was correctly calculated for about 70% of the input images. This is due to both incorrectly identified characters and missed characters, during the calculation of the best fit function to obtain the angle. Finding a deskew angle that fits the whole word was generally not possible, because each character had its own skew. The deskew performance was lower, 20%, due to this fact.

The text extraction was not very successful. Despite a better pre-processing can be achieved, currently the OCR library used is also not the most adequate since, it is trained for “normal” non-skewed text. Since there wasn’t a single correct answer. For most images, in average only 2-4 characters were correctly recognized. The characters, even after pre-processing, are too deformed to be recognized by the library. Note that this is an open-source library focused mainly for texts with no need for pre-processing, such as scanned PDF files. No proprietary libraries were used in this module.

Since improving the pre-processing even more is a very difficult and time-consuming task, future work is focused on improving the text extraction. This is done by creating a library from scratch, trained to recognized characters from texts on vessels. Gathering a collection of multiple images of the letter/number retrieved from vessel’s registrations and training a HOG cascade classifier for each of the characters is one possibility for greatly improve this task’s performance.

B. Boarded System

The boarded system, due to the VMS’s lack of processing, requires a light algorithm capable of counting and tracking fishing gear. The algorithm must not be too sensible and it is admissible some missed gear, as long as the algorithm can distinguish an attempt to fish from an intentional or non-intentional attempt to deceive the system.

The algorithm previously as described, applies the principles of pixel-based detection and achieves a good performance in exchange for a small processing time and resources.

A camera was installed in stern of the vessel “Noruega”, and acquired footage of over a week during IPMA’s research cruise on the Atlantic sea. From the footage acquired, around 160 minutes contain fishing nets used during trawling and were used to test the algorithm. From the 160 minutes of footage, it was created a dataset composed of 160 images, each image corresponding to a frame of the footage at a rate of one frame per minute. To test the robustness an additional dataset containing around 200 images with no fishing gear was added.

In the test environment, only 3 false positives were detected out of the 160 positive images. Out of the 200 false images, 27 were incorrectly classified as containing fishing gears. The overall performance is around 85% which is a great result.

These results were further improved with a second function of the DLL: data analysis. This function analysis the detections along the time and compensates errors. In other words, if a gear is detected periodically and during an analysis it was not detected, then the data analysis can identify this situation as being a

singularity and discard the occurrence. Similarly, if the system detects once a gear in a large period of time, then it can be considered as a false positive and also discard the occurrence. With the introduction of this function, the results amounted a performance of 100%.

Currently the VMS system had a delay in its release and is still in development. Because of this issue, there are no field tests, on the hand the test footage was composed of videos acquired in a vessel where the field tests will occur. This means that only the processing unit and will only affect the algorithm's execution times but not its performance.

Figure 43 – Example of the input (left) and output (right) of the boarded system

VI. CONCLUSION AND FUTURE WORK

The SeaItAll project is a vast platform compose of a wide diversity of interconnected systems with the final goal of reducing Illegal, unreported and unregulated (IUU) fishing. Most of these systems are still on development and are outside the scope of this thesis. The main focus is on two systems: the detection of vessels entering or exiting the harbours, non-boarded system, and logging each fishing activity while on sea, boarded system.

The non-boarded system must be able to detect moving vessels, photograph it and extract all the text visible. As such, it was divided in 3 modules: motion detection (motion detection module), text area detection (Region Of Interest analysis module) and text recognition (Optical Character Recognition module). The boarded system must detect fishing gear on a photograph provided by the calling software.

As discussed throughout this paper and attending the requirements, the non-boarded system requires both a pixel-based approach and an object-based classifier. The first approach is required to enable the vessel detection module with the capability of triggering the photographic camera whenever a vessel passes through the canal. This was the faster option, since the camera's field of view and frame rate were not able to ensure the detection of the vessel using a classifier. The object-based classifier is used to locate the vessel's ROI and recognizing its characters. This approach typically presents better results based on a great amount of data, instead of a fixed set of rules designed by a programmer. Additionally, an OCR engine is required to identify each character.

About the first module, the vessel detection module, there is a trade-off between noise resistance and sensitivity, as a very sensitive approach have low noise resistance, and activates the camera with non-relevant changes such as lighting, atmospheric conditions or other factors, such as birds. On the other hand, if the solution is too resistant to noise, it ignores the presence of small vessels, moving too fast or that are too far away from the camera. The approach 4 (the more complex pixel-based approach) was selected due to its performance in both lab and real-life scenarios. Finding the perfect value for the threshold requires trial and error. Plus, changes in the environment may require adjustments in the value. Overall, this solution is far from perfection and is far from being an autonomous solution that could easily be installed without needing adjustments. Other alternatives developed solve these issues but its performances are not enough for the specifications required. Further work is needed to ensure a better performance, for instance, combining the approach used, with a low threshold value, and verifying the result with an additional module, composed of a classifier trained to recognize vessels. Such additional module was an option tried and it was concluded that a very large dataset was required, as well as a great amount of time to select the vessels in each image.

The ROI analysis module, i.e. the second module, achieves a good performance with very low number of false positives and detects all legible texts in the test dataset. The illegible texts are generally reflecting too much light and are only perceived due to the location they occupy in the vessel. This shows another aspect to improve in this module: the capability to recognise a determined region as very likely to contain text. Despite such good results, it's very probable to miss some texts in different environments not yet tested. Further work requires a wider training dataset with more variety and in different conditions.

The third module, OCR module, still requires more iterations for it to present proper results. As is, the module calculates the correct rotation most of the times. The segmentation of the characters and deskew calculation still have limited performances and more work needs to be put up to ensure a better segmentation and deskew.

Finally, the character recognition uses the tesseract library, generally used for text with little to no distortions, and as such the performance is very poor. Future work consists in creating a new OCR library trained with distorted characters, contemplating various conditions such as the ones faced by this module. More research is needed to create a better segmenting algorithm. Also, as suggested during early stages of the SeaItAll project planning, a database with the name of the vessels belonging to the harbour/country and a comparison/correction algorithm would significantly increase the performance of the module.

As for the boarded system, multiple trained object-based classifiers would be required to identify each of the supported fishing gear. Since generally the vessel uses only one type of gear, the approach where each of the classifiers are tested against the images obtained is the fastest and the one that requires less processing. This would require a classifier for each type of fishing gear. However, considering the amount of time and data required to create a good classifier, an alternative was required. Using a pixel-based detection approach proved to work well, plus being considerably simpler.

During the elaboration of this dissertation some difficulties arose:

- One of the main difficulties is related to the programming languages since it was specified which languages would have to be used, in order to fit the requirements. To prepare the project, an advanced level of knowledge in any of the languages (.NET, C ++, C # and Matlab) was required.
- As mentioned earlier, the lack of standards was probably one of the main issues during this work. The standards are so vague that it originates a wide variety of different scenarios. Another related issue is the lack of inspection, since during this work it was noted most vessels exceed the maximum allowed speed in harbour, which was also the maximum speed allowed by the hardware, and the character sizes, places and colours were also often illegal. Creating rigid standards and inspections would certainly ease the detection of vessel registrations.

- The architecture of the systems has also been an issue, since the use of DLLs limits the performance and, therefore, limits the applicable approaches, mainly due to the fact that a DLL does not store objects in memory and as such it becomes difficult to create an historic of objet changes (like for example creating a kalman filter). Moreover, since two different operating systems were used, it was necessary to adapt certain characteristics in the code, so as to adapt to the new environment.
- Some difficulties were felt while using the libraries (opencv and tesseract), such as outdated, and therefore confusing, information in forums/documentation.
- The lack of data to train classifiers has limited the use of object-based classification and tests. A lot of both negative and positive images are required to create a good classifier. Finding positive images was particularly hard because of the specific type of vessels used and minimum quality standards. This was most felt in the boarded system which had very little data to train and test.
- Segmenting the positive images is a very time-consuming task. It takes days to draw a region of interest around every vessel in each of the positive images.
- Limitations in hardware (e.g. minimum and maximum distance of image capture) and terrain characteristics (e.g. camera positioned in front of a zone with background movement) are variables that had to be considered and consisted in an obstacle to the implementation of the systems. Project budgeting and timing may have essentially limited hardware and field testing, as well.
- Last and most important, the lack of experience and knowledge in the area was an issue, especially during early stages.

Overall, the experience was gratifying, because throughout the development of the modules I was faced with small details that apparently seem easy to get around but in reality, are difficult to solve taking into account the current technology. The experience and the research for new solutions are decisive factors in solving the various problems that arose, making this work a bridge between theory and practice.

VII. REFERENCES

[1] Moeslund, T. B., Hilton, A., Kruger, V., & Sigal, L. (2011). Visual Analysis of Humans: Looking at People. https://doi.org/10.1007/978-0-85729-997-0

[2] Mahadevan, V., Li, W., Bhalodia, V., & Vasconcelos, N. (2010). Anomaly detection in crowded scenes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1975–1981. https://doi.org/10.1109/CVPR.2010.5539872

[3] Mehran, R., Oyama, a, & Shah, M. (2009). Abnormal Crowd Behaviour Detection using Social Force Model. IEEE Conference on Computer Vision and Pattern Recognition, (2), 935–942.

[4] Abuarafah, A. G., Khozium, M. O., & AbdRabou, and E. (2012). Real-time crowd monitoring using infrared thermal video sequences. Journal of American Science, 8(3). Retrieved from http://www.tcmcore.org/en/pdf/133140.pdf on June 2017

[5] Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, I, 886–893. https://doi.org/10.1109/CVPR.2005.177

[6] Jalled, F., & Voronkov, I. (2016). Object Detection using Image Processing, 1–6. Retrieved from http://arxiv.org/abs/1611.07791

[7] Perez, M. (2005). Vision-Based Pedestrian Detection for Driving Assistance, 9. Retrieved from http://users.ece.utexas.edu/~bevans/courses/ee381k/projects/spring05/perez/LitSurveyReport.pdf http://users.ece.utexas.edu/~bevans/courses/ee381k/projects/spring05/perez/LitSurveyReport.pdf

[8] Lu, S., Tsechpenakis, G., Metaxas, D. N., Jensen, M. L., & Kruse, J. (2005). Blob analysis of the head and hands: A method for deception detection. Proceedings of the 38th Annual Hawaii International Conference on System Sciences (HICSS '05), 1–10. https://doi.org/10.1109/HICSS.2005.122

[9] Gao, Y., & Mas, J. (2008). A comparison of the performance of pixel based and object based classifications over images with various spatial resolutions. Online Journal of Earth Sciences, 2(8701), 27–35. https://doi.org/doi:ojesci.2008.27.35

[10] Haj, M. A. & Fernández, C. & Xiong, Z & Huerta, I. & Gonzàlez, J. & Roca, X. (2013). Visual Analysis of Humans. Journal of Chemical Information and Modeling, 53, 1689–1699. https://doi.org/10.1017/CBO9781107415324.004

[11] Simard, P. Y., Steinkraus, D., & Platt, J. C. (n.d.). Best practices for convolutional neural networks applied to visual document analysis. In Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. (Vol. 1, pp. 958–963). IEEE Comput. Soc. https://doi.org/10.1109/ICDAR.2003.1227801

[12] Angelova, A., Krizhevsky, A., View, M., View, M., Vanhoucke, V., Ogale, A., & Ferguson, D. (2015). Real-Time Pedestrian Detection With Deep Network Cascades. Bmvc2015, 1–12. https://doi.org/10.5244/C.29.32

[13] Blaha, F. (2016). Electronic monitoring on fishing vessels — Francisco Blaha. Retrieved December 2, 2016, from http://www.franciscoblaha.info/blog/2016/3/23/cud5ifuynu0cuwxxk3wa9v0s1lz3ch

[14] European Commission. (n.d.). Scope of the control system | Fisheries. Retrieved December 1, 2016, from https://ec.europa.eu/fisheries/cfp/control/scope_of_the_control_system_en

[15] McElderry, H. (2006). At-Sea Observing Using Video-Based Electronic Monitoring. ICES Annual Science Conference, (July).

[16] European Commission. (n.d.). Vessel monitoring system (VMS) | Fisheries. Retrieved December 1, 2016, from https://ec.europa.eu/fisheries/cfp/control/technologies/vms_en

[17] Witt, M. J., Godley, B. J., Pauly, D., Christensen, V., Guenette, S., Pitcher, T., Mountain, D. (2007). A Step Towards Seascape Scale Conservation: Using Vessel Monitoring Systems (VMS) to Map Fishing Activity. PLoS ONE, 2(10), e1111. https://doi.org/10.1371/journal.pone.0001111

[18] Jennings, S., & Lee, J. (2012). Defining fishing grounds with vessel monitoring system data — Defining fishing grounds with vessel monitoring system data — Supplementary Data, 69, 51–63. Retrieved from http://icesjms.oxfordjournals.org/content/69/1/51/suppl/DC1

[19] Heikkilä, M., & Pietikäinen, M. (2006). A texture-based method for modeling the background and detecting moving objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(4), 657–662. https://doi.org/10.1109/TPAMI.2006.68

[20] Mangi, S. C., Dolder, P. J., Catchpole, T. L., Rodmell, D., & de Rozarieux, N. (2013). Approaches to fully documented fisheries: Practical issues and stakeholder perceptions. Fish and Fisheries, 426–452. https://doi.org/10.1111/faf.12065

[21] Lienhart, R., Kuranov, A., & Pisarevsky, V. (2003). Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Proceedings of the 25th DAGM Pattern Recognition Symposium, 297–304. https://doi.org/10.1007/978-3-540-45243-0_39

[22] Google. (n.d.). Nazaré. n.d. Retrieved from https://goo.gl/pn9VFO

[23] Understanding Features — OpenCV 3.0.0-dev documentation. (n.d.). Retrieved January 16, 2017, from http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_features_meaning/py_features_meaning.html#features-meaning

[24] SimpleCV. (n.d.). Retrieved January 16, 2017, from http://simplecv.org/

[25] Python Computer Vision Framework. (n.d.). Retrieved January 16, 2017, from http://pycvf.sourceforge.net/

[26] FastCV Computer Vision SDK - Qualcomm Developer Network. (n.d.). Retrieved January 16, 2017, from https://developer.qualcomm.com/software/fastcv-sdk

[27] Khoros. (n.d.). Retrieved January 16, 2017, from http://www.agocg.ac.uk/reports/visual/vissyst/dogbo_45.htm

[28] xpcv – Cross-Platform Computer Vision Framework. (n.d.). Retrieved January 16, 2017, from https://www.xpcv.de/features/9/

VIII. ANNEX