



ISEL – Instituto Superior de Engenharia de Lisboa

ADEETC – Área Departamental de Engenharia de  
Eletrónica e Telecomunicações e de Computadores

---

LEIM

Licenciatura em Engenharia Informática e Multimédia  
Unidade Curricular de Projeto

---

## Analítica de vídeo para drones em missões de busca e salvamento



Michael Madeira (41032)

João Pedro Santos (41875)

*Orientador (es)*

---

*Professor Pedro Mendes Jorge*

*Professor António Serrador*

---

*Julho, 2018*



# Resumo

Este projeto tem como objetivo o desenvolvimento de um algoritmo de detecção de naufragos em tempo real num ambiente de zona costeira, utilizando um veículo aéreo não tripulado. Para esta detecção recorre-se ao processamento de sequência de imagens, extraíndo características relevantes e classificando-as em entidades/conceitos relevantes para o objetivo deste projeto. São utilizadas técnicas baseadas em extração de características, segmentação de imagem baseada em textura e aprendizagem automática.

Os resultados obtidos pelo algoritmo de detecção serão avaliados utilizando métricas, como por exemplo, *precision&recall*, *accuracy* e *F-score*.

Para desenvolver este projeto foi utilizada a linguagem de programação Python, assim como algumas bibliotecas (Numpy, OpenCV, Scikit-learn entre outras).

Sendo este processamento inicialmente limitado, pois o drone a ser utilizado será um DJI Inspire 1, que pelas suas características não é OpenSDK, consequentemente não iremos processar essas imagens em tempo real, mas sim após a captação das mesmas durante o voo.

**Palavras-Chave:** Detecção de Naufragos; Drone; Veículo Aéreo Não Tripulado; Processamento de Imagem; Segmentação de textura; Extração de Características; Aprendizagem Automática.



# Abstract

This project has the objective of developing an algorithm capable of detecting a victim in a coastal zone environment, using an unmanned aerial vehicle in real time. For this detection we have sequence of images being processed, extracting relevant features of it and classifies it effectively and precisely.

There will be techniques like Texture Segmentation, Feature Extraction and Automatic Learning.

The results obtained by the automatic detection algorithm were evaluated using evaluation metrics (precision, recall, accuracy and F-score).

To develop this project was used the programming language Python, as well as some libraries (Numpy, OpenCV, Scikit-learn among others).

Initially this process will be limited because Inspire 1 isn't OpenSDK, so we will not be able to process the images in real time, just after the flight.

**KeyWords:** Detection of Castaways; Drone; Unmanned Aerial Vehicle; Image Segmentation; Texture Segmentation; Extraction of Features; Machine Learning.



# Agradecimentos

Gostaríamos de aproveitar este espaço para agradecer a todas as pessoas que contribuíram para a elaboração deste projeto.

Aos Professores Pedro Mendes Jorge e António Serrador, pela disponibilidade, por todo o conhecimento que nos foi transmitido e pela orientação ao longo deste semestre.

O Professor Paulo Trigo, da Unidade Curricular de Projeto, pelos conhecimentos e organização transmitida nas aulas.

Ao Grupo Microsegur, pelo apoio e pela disponibilidade dos meios materiais para o desenvolvimento do trabalho.

Aos nossos amigos por nos darem motivação, opiniões, sugestões e críticas para melhoramento do projeto.

Por fim às nossas famílias por nos apoiarem desde sempre nos bons e maus momentos deste projeto.





*Agradecimento as nossas famílias*

*Pela paciência e pela*

*fé imposta em nós*



# Índice

Resumo .....	i
Abstract.....	iii
Agradecimentos.....	v
Índice.....	ix
Lista de Acrónimos .....	xi
Lista de Tabelas.....	xiii
Lista de Figuras .....	xv
Introdução .....	17
Trabalho Relacionado.....	19
2.1 Aplicações .....	19
2.2 Estado da arte .....	19
2.2.1 Segmentação por textura .....	19
2.2.2 Classificação .....	20
Modelo Proposto.....	23
3.1 Requisitos .....	23
3.1.1 Caracterização Geral .....	23
3.1.2 Caracterização Pormenorizada .....	24
3.1.3 Casos de Utilização.....	26
3.2 Abordagem.....	28
3.2.1 Arquitetura .....	28
Implementação do Modelo .....	29
4.1 Pré-processamento .....	31
4.1.1 Plano Gray .....	31
4.1.2 Plano Lab .....	32
4.1.3 Sub-Amostragem .....	33
4.1.4 Divisão em blocos .....	33
4.1.5 Teste à sub-amostragem e divisão em blocos .....	34
4.2 Extração de Características.....	37

4.2.1 Decisão da distância e ângulo a utilizar .....	38
4.3 Classificação .....	39
4.3.1 Classificação SEAGULL .....	40
4.3.2 Classificação Final .....	40
4.4 Métodos Combinados para melhor identificação.....	42
4.4.1 Técnica de segmentação por cor adaptada .....	42
4.4.2 <i>Tracking</i> .....	42
Validação e Testes.....	47
5.1 Base de dados .....	47
5.2 Métodos de Avaliação .....	48
5.3 Testes Efetuados .....	49
Conclusões e Trabalho Futuro .....	55
Bibliografia.....	57

# Lista de Acrónimos

<b>GLCM</b>	<i>Gray Level Co-Occurrence Matrix</i>
<b>SVM</b>	<i>Support Vector Machine</i>
<b>RGB</b>	<i>Red-Green-Blue</i>
<b>TP</b>	<i>True Positive</i>
<b>FP</b>	<i>False Positive</i>
<b>TN</b>	<i>True Negative</i>
<b>FN</b>	<i>False Negative</i>
<b>GCS</b>	<i>Ground Control Station</i>
<b>AAN</b>	Autoridade AeroNáutica Nacional
<b>ICNF</b>	Instituto de Conservação da Natureza e das Florestas
<b>ANAC</b>	Agência Nacional de Aviação Civil



# Lista de Tabelas

Tabela 1 - Exemplos de métodos utilizados para segmentação por textura	20
Tabela 2 - Funções gerais do sistema	24
Tabela 3 - Funções do Sistema - Pré-Processamento	24
Tabela 4 - Funções do Sistema - Extração de características	24
Tabela 5 - Funções do Sistema - Classificação	24
Tabela 6 - Funções do Sistema - Pós-processamento	25
Tabela 7 - Funções do Sistema - Sinalização ao operador	25
Tabela 8 - Atributos do Sistema	25
Tabela 9 - Atores nos Processos do Sistema	26
Tabela 10 - Caso de Utilização do processamento das imagens e detecção do naufrago	27
Tabela 11 - Casos de Utilização do reconhecimento do operador, se é pessoa ou não pessoa e o alerta as entidades	27
Tabela 12 – Testes das várias características/features extraídas no GLCM nos vários Planos	32
Tabela 13 - Matriz de coocorrência	37
Tabela 14 - Ângulos e respectivos contrastes da Figura 12	38
Tabela 15 - Novas características diferenciadoras	40





# Lista de Figuras

Figura 1-Evolução de técnicas de segmentação por textura [Li Liu, 2018]	20
Figura 2 - Casos de utilização	26
Figura 3 - Arquitetura do Sistema	28
Figura 4 – Diagrama de Blocos da abordagem implementada. i) Imagem Original; ii) Imagem Gray; iii) Imagem dividida em blocos $M \times N$ ; iv) Classificação das características extraídas do GLCM; v) Máscara resultante da classificação; vi) Imagem com a identificação, do resultado obtido da máscara	30
Figura 5 - Imagem original e imagem gray, respetivamente	31
Figura 6-Imagem original e imagem do plano ‘b’, respetivamente	32
Figura 7 - Frame 3580 em plano Gray dividida por 8, com interpolação Cubic, em blocos de $8 \times 8$ na janela do canto superior esquerdo; Representação 3D dos 3 pontos, pessoa, espuma e mar, no canto superior direito com as features Media, Desvio e Contraste (a), (b) e (c).	34
Figura 8 - bloco $8 \times 8$ e imagem/2 (a); bloco $16 \times 16$ e imagem/2 (b); bloco $8 \times 8$ e imagem/4 (c); bloco $16 \times 16$ e imagem/4 (d); blocos $8 \times 8$ e imagem/4 (e); bloco $8 \times 8$ e imagem/8 (f).	35
Figura 9 - Frame 4514 em plano Gray dividida por 8, com interpolação Cubic, em blocos de $8 \times 8$ na janela do canto superior esquerdo; Representação 3d dos 3 pontos, pessoa, espuma e mar, no canto superior direito com as features Media, Desvio e Contraste (a), (b) e (c)	36
Figura 10 - Blocos de níveis de cinzento e bloco de etiquetas de acordo com os níveis de cinzentos, respetivamente	37
Figura 11 – Exemplo de grupo de pixéis, de níveis de cinzento, com maior número de ocorrências na imagem	37
Figura 12 - Bloco de $M \times N$	38
Figura 13 - Classificador SVM	39
Figura 14 - Imagens de Treino - Frame 3580 e Frame 4512, respetivamente	40
Figura 15 - Features extraídas	41
Figura 16 - Diagrama de estados do Tracking	43
Figura 17 - Frames mal classificadas	44
Figura 18 - Sequência das mesmas frames, que a Figura 16, mas com a técnica de segmentação de cor adaptada	44
Figura 19 - Sequência de frames com o tracking, onde se vê que na imagem b) a pessoa já é identificada	45
Figura 20- 4 imagens de 4 sequências de teste do vídeo de salvamento	47
Figura 21- Imagem da sequência de imagens do vídeo do surfista	48
Figura 22 - Matriz de confusão padrão	49
Figura 23 - Ground-thruth (a), Classificador (b), Ajuste (c), Tracking (d)	50

Figura 24 - Matriz de co-ocorrência das imagens (b) e (d)	50
Figura 25 - Matriz de co-ocorrência das imagens (b) e (d)	51
Figura 26 - Ground-thruth (a), Classificador (b), c/Ajuste (c),c/ Tracking (d)	51
Figura 27 - Ground-thruth (a), Classificador (b), c/Ajuste (c), c/Tracking (d)	52
Figura 28 - Matriz de co-ocorrência das imagens (b) e (d)	52
Figura 29 - Matriz de co-ocorrência das imagens (b) e (d)	53
Figura 30 - Ground-thruth (a), Classificador (b), c/Ajuste (c), c/Tracking (d)	53
Figura 31- Imagem 614 classificada com o classificador treinado com a base de dados do video do salvamento	54

# Capítulo 1

## Introdução

Este projeto tem como objetivo identificar naufragos que estejam em perigo, num ambiente de zona costeira utilizando um *drone*. Pretende-se que seja desenvolvido um sistema semiautónomo capaz de alertar e facilitar as entidades especializadas, a agir de uma forma imediata, a fim de ajudarem as vítimas que se encontram em perigo. Para isso têm que ser abordadas questões legislativas e de planificação da missão de voo, *know how* que é necessário para perceber quais as condições a respeitar no uso do *drone* para este tipo de missões.

O *drone* sendo um veículo aéreo não tripulado, é uma solução prática, rápida, barata e que exige poucos recursos humanos e de comunicação. Estas características tornam o *drone* um meio muito útil, em missões de busca e salvamento pois permitem uma redução de custos e recursos, das entidades competentes envolvidas nestas operações.

O objetivo principal deste projeto é o desenvolvimento de um algoritmo que seja capaz de processar, em tempo real, imagens captadas pelo *drone*, e que não tenha um poder computacional elevado. Devido às variadas condições que poderá haver na área de operações, a transmissão de grandes sequências de imagem para a *ground station* não será a melhor abordagem, pelo que serão escolhidas pequenas sequências relevantes para dar o alerta ao operador do *drone*, e que serão validadas pelo mesmo.

De forma a patrulhar a linha de costa eficientemente, é importante ter em atenção as condições adversas que possam dificultar a identificação de uma possível vítima. Condições adversas do estado do mar, espuma e reflexos da luz solar condicionam o desempenho do algoritmo de deteção. Irão ser tomadas medidas para ir ao encontro do objetivo principal e destas adversidades. São utilizadas características de textura, como por exemplo, as matrizes de co-ocorrência (GLCM – *Gray-Level-Co-Occurrence Matrix*) de onde se extrai as melhores características para segmentar o fundo da imagem a nível da textura, a média e o desvio do plano Gray para segmentar o objeto. A respeito do custo computacional, existem dois pontos diferenciadores que nos obriga a tratar as imagens de maneiras distintas, o Zoom e a Resolução.

Consoante o Zoom e a Resolução da imagem, dividimos a imagem por blocos  $M \times N$ , ajustado a estes parâmetros. As características retiradas deste pré-processamento serão utilizadas para treinar o modelo de aprendizagem supervisionado (SVM – *Support Vector Machine*).

A base de dados a ser utilizada será inicialmente, imagens e vídeos da SEAGULL [Miguel Griné., 2014], onde irão ser colocadas à prova, todas as características a utilizar e todos os processos do classificador SVM. Seguindo-se um vídeo de salvamento aquático, que determinará os nossos resultados finais.

Este trabalho foi projetado para detetar alvos específicos, mas pode ser usado para detetar outro tipo de objetos, como barcos ou derrames, tendo de ser providenciada outra base de dados desse tipo de objetos. Ou seja, só é necessário um conjunto de imagens e características para treinar o modelo, capazes de identificarem bem o tipo de objeto alvo.

No pós-processamento temos em consideração a variável temporal, em termos de estabelecer alguma coerência entre *frames* através do método de seguimento (*tracking*). Esta última contribuição do pós-processamento fecha todas as estratégias para melhorar o algoritmo ao máximo, onde a contribuição de todas as técnicas de pré-processamento utilizadas também entram nessa mesma estratégia.

A estrutura do relatório está de acordo com o processo de desenvolvimento *Unified Process*:

- **Trabalho Relacionado (Capítulo 2)** – neste capítulo apresenta-se o estado da arte, fundamenta-se e contextualiza-se as técnicas selecionadas para a realização deste trabalho;
- **Modelo Proposto (Capítulo 3)** – neste capítulo é feita uma análise dos requisitos do sistema, identificando quais as funções, atributos do sistema e casos de utilização;
- **Implementação (Capítulo 4)** – neste capítulo descrevemos de forma rigorosa e detalhada todos os procedimentos propostos para o desenvolvimento do projeto;
- **Validação e Testes (Capítulo 5)** – neste capítulo apresentamos os testes e as validações extraídas ao longo do projeto;
- **Conclusões e Trabalho Futuro (Capítulo 6)** – neste capítulo são apresentadas as conclusões retiradas do trabalho realizado e quais os próximos objetivos para implementar no projeto.

O que nos motivou fazer este projeto, foi a vontade de desenvolver um sistema que atualmente não encontra em uso a nível nacional, por ser um mercado ainda em desenvolvimento.

# Capítulo 2

## Trabalho Relacionado

De forma a percebermos o que já está desenvolvido nesta área começámos por estudar técnicas de segmentação de imagens, para dividir o que é mar e não mar. Dado que o mar é considerado um padrão, resolvemos consultar estudos que indiquem as melhores técnicas para segmentação por texturas. Para o nosso problema a melhor solução estudada é o GLCM.

Identificar também aplicações que sejam semelhantes ao nosso projeto, de maneira a poder aplicarmos o nosso algoritmo.

### 2.1 Aplicações

Posto que a tecnologia dos *drones* ainda é muito embrionária, a nível internacional e nacional, não existem muitas aplicações relacionadas com o nosso projeto. Porém, podem-se destacar algumas áreas relacionadas:

- Detecção de embarcações com avarias, em alto mar;
- Detecção de resíduos tóxicos, derramados no mar;
- Controlo de pescas clandestinas;
- Detecção de embarcações insufláveis, com imigrantes a bordo.

### 2.2 Estado da arte

#### 2.2.1 Segmentação por textura

Com a evolução das tecnologias, a nível computacional, e das necessidades que vão surgindo a nível de reconhecimento de padrões, pretende-se que os algoritmos até agora desenvolvidos acompanhem o ritmo, e que não sejam esquecidos dos que anteriormente trouxeram bons resultados em diferentes áreas, como análise de imagens médicas, qualidade de inspeção, análise facial, reconhecimento de objetos e tantas outras, Figura 1.

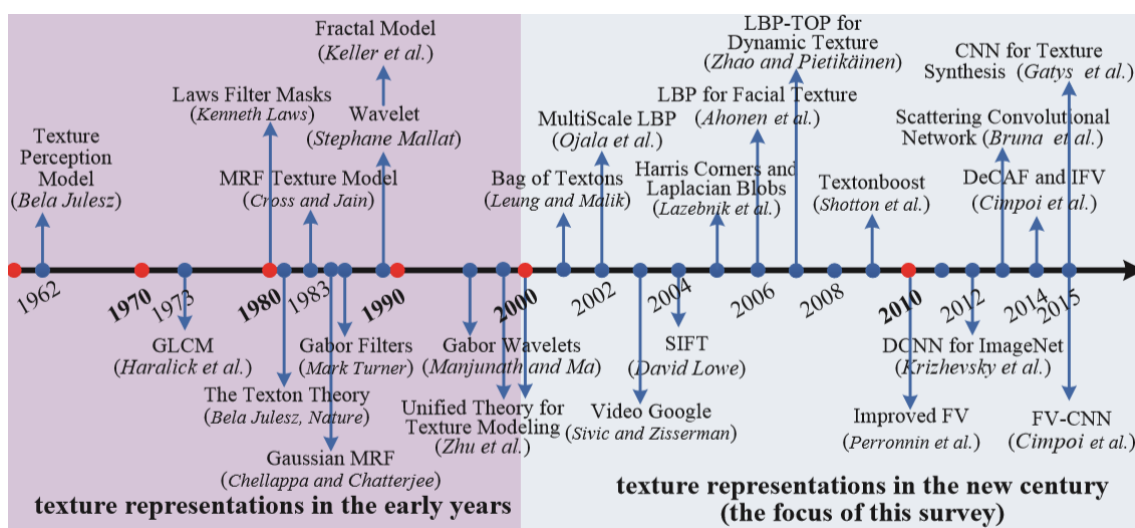


Figura 1-Evolução de técnicas de segmentação por textura [Li Liu, 2018]

O nosso problema insere-se na área de reconhecimento de objetos, com a detecção de possíveis naufragos. Numa primeira fase, pretendemos salientar regiões que para nós, sejam *background*. O nosso *background*, é o mesmo da detecção de barcos em alto mar [Miguel Griné., 2014], que tem abordagens que nos seriam úteis, de onde retiramos que a melhor característica para salientar o que é mar do que não é, seria o contraste da imagem.

Para extrair esta característica, identificamos os métodos mais utilizados para a segmentação de textura, com percentagens de acerto típicas.

Métodos	Vantagens	Desvantagens	Percentagem de acerto
GLCM	Melhor informação	Maior custo computacional	75
LAWS	Pouca informação	Maior custo computacional	62
Gabor Filter	Pouca informação	Menor custo computacional	70

Tabela 1 - Exemplos de métodos utilizados para segmentação por textura

### 2.2.2 Classificação

A ideia principal é separar todas as características em clusters, e conseguir identificar quais as melhores componentes para termos uma melhor distribuição espacial desses clusters. Decidimos utilizar o classificador supervisionado SVM linear, que será treinado manualmente com a nossa base de dados e é um excelente classificador binário. Temos razão para acreditar que o SVM é um bom classificador para

---

identificar pessoas no mar, também pela razão que nos chamou à atenção no estudo [Miguel Griné., 2014], onde também é utilizado para classificar barcos.





# Capítulo 3

## Modelo Proposto

Neste capítulo é feita uma análise de requisito, mais pormenorizada. Para isso é necessário, identificar quais os requisitos do sistema, organizar os mesmos em relação aos casos de utilização e por último ordenar os casos de utilização.

### 3.1 Requisitos

Os requisitos do sistema apresentam-se subdivididos:

- **Caracterização Geral** – identifica quais os objetivos do projeto, os clientes a que se destina e quais as metas a alcançar;
- **Caracterização Pormenorizada** – menciona as funções e os atributos do sistema.

#### 3.1.1 Caracterização Geral

##### **Objetivos**

Este projeto tem como objetivos o desenvolvimento de um sistema para detetar náufragos, no mar. Também tem como objetivo, solucionar um meio para ajudar entidades como a Polícia Marítima, em missões de busca e salvamento.

Dado como desaparecida uma pessoa, é lançado o drone. Detetado o náufrago é enviado um alerta para o operador, que informa as autoridades da ocorrência.

##### **Clientes**

Os clientes destinados para o projeto serão as entidades indicadas para Proteção, Vigilância e Segurança do Domínio Público Marítimo.

##### **Metas a Alcançar**

- Processamento rápido e eficiente, posto que está em perigo a vida das pessoas;
- Anular os falsos positivos e minimizar falsos negativos, no menor tempo possível.

### 3.1.2 Caracterização Pormenorizada

#### Funções do Sistema

As funções do sistema representam as suas funcionalidades, que devem ser hierarquizadas de modo a definir as prioridades das mesmas. A maioria das funções existentes são categorizadas como “Invisível”, de modo a que o utilizador não possa ver. Exceto a função de sinalizar no ecrã e deteção do naufrago, que é visível.

Na Tabela 2, são identificadas as funções gerais do sistema.

Ref.	Função
R1	Pré-processamento do vídeo
R2	Extração de características
R3	Classificação
R4	Pós-processamento
R5	Sinalização dada ao operador

*Tabela 2 - Funções gerais do sistema*

Na função de pré-processamento, são utilizados métodos para reduzir o custo computacional e requisitos que são necessários para a função R2, antes de se efetuar a deteção, Tabela 3.

Ref.	Função	Categoria
R1.1	Redução da resolução da imagem	Invisível
R1.2	Transformação da imagem RGB para <i>Gray</i>	Invisível
R1.3	Divisão da imagem em blocos NxN	Invisível

*Tabela 3 - Funções do Sistema - Pré-Processamento*

No tópico de extração de características, mostramos quais os métodos utilizados para a segmentação por textura, como podemos verificar na Tabela 4.

Ref.	Função	Categoria
R2.1	Cálculo da Matriz de Coocorrência	Invisível
R2.2	Extração de propriedades	Invisível

*Tabela 4 - Funções do Sistema - Extração de características*

Na Tabela 5, encontramos as funções para a classificação.

Ref.	Função	Categoria
R3.1	Classificação do que é mar e não mar	Invisível

*Tabela 5 - Funções do Sistema - Classificação*

Depois de feita a classificação assinalamos, na imagem original a localização do naufrago. Na Tabela 6, encontramos as funções implementadas, para efetuar o que foi mencionado inicialmente.

Ref.	Função	Categoria
R4.1	Classificação do que é pessoa e não pessoa	Invisível
R4.2	<i>Tracking</i> – reduzir os falsos negativos	Invisível
R4.3	Construímos a máscara com o resultado	Invisível
R4.4	Identificamos os locais de interesse na imagem original	Invisível

*Tabela 6 - Funções do Sistema - Pós-processamento*

Por último, na Tabela 7, apresentamos a função que tem o objetivo de sinalizar o operador.

Ref.	Função	Categoria
R5.1	Identificar no ecrã onde está a pessoa	Visível
R5.2	Alertar as entidades	Visível

*Tabela 7 - Funções do Sistema - Sinalização ao operador*

### Atributos do Sistema

Nesta secção são apresentadas as características do sistema, que são qualidades não-funcionais, como se pode visualizar na Tabela 8.

Atributo	Detalhe/Restrição de Fronteira	Categoria
Interação Homem-Máquina	Gráfica Identificar o naufrago	Obrigatória
Tempo de Resposta	Menos de 1 segundo	Desejável
Plataformas	<i>Python, OpenCV, Scikit-Learn, Scikit-Image, Numpy</i>	Desejável

*Tabela 8 - Atributos do Sistema*

### 3.1.3 Casos de Utilização

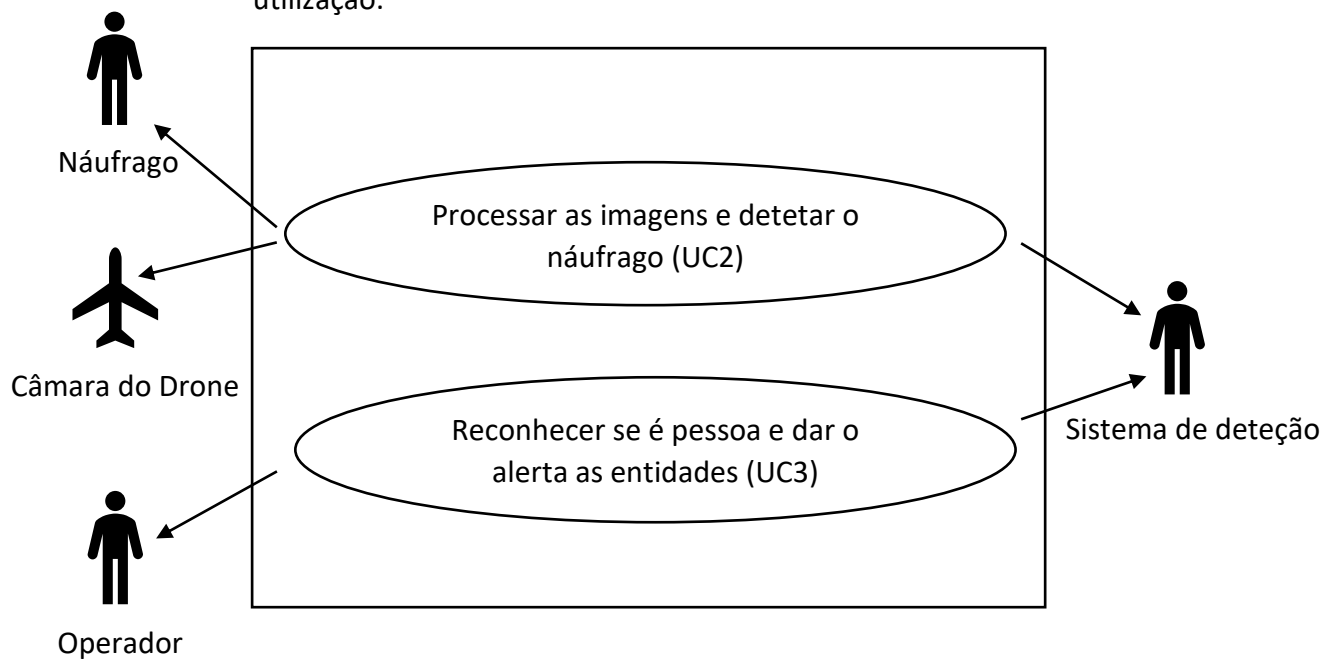
#### Atores

Na Tabela 9 podemos observar os atores dos processos existentes no sistema.

Actor	Processo
Náufrago	?
Câmara do <i>Drone</i>	Captar as imagens
Sistema de deteção	Processar as imagens e detetar o náufrago
Operador	Reconhecer se é pessoa e dar o alerta as entidades

*Tabela 9 - Atores nos Processos do Sistema*

Na Figura 2, está descrita a relação entre os atores e os casos de utilização.



*Figura 2 - Casos de utilização*

Nas próximas Tabela 10 e 11 apresentamos os casos de utilização, representados na Figura 2.

Caso de Utilização UC1	
<b>Nome</b>	Processar as imagens e detetar o náufrago
<b>Resumo</b>	As imagens captadas pelo <i>drone</i> são processamento e identificam onde está o náufrago
<b>Referência</b>	R1.1; R1.2; R1.3; R2.1; R2.2; R3.1; R4.1; R4.2; R4,3; R4.4

*Tabela 10 - Caso de Utilização do processamento das imagens e deteção do náufrago*

Caso de Utilização UC2	
<b>Nome</b>	Reconhecer se é pessoa e dar o alerta as entidades
<b>Resumo</b>	O operador verifica no ecrã, onde está identificado o náufrago, se é pessoa e dá o alerta.
<b>Referência</b>	R5.1; R5.2

*Tabela 11 - Casos de Utilização do reconhecimento do operador, se é pessoa ou não pessoa e o alerta as entidades*

## 3.2 Abordagem

### 3.2.1 Arquitetura



Figura 3 - Arquitetura do Sistema

Neste projeto foi necessário aplicar o conhecimento relacionado com áreas estudadas ao longo dos 3 anos de licenciatura, como por exemplo: Programação, Processamento de Imagem e Aprendizagem Automática. Foi necessário decidir quais as tecnologias relacionadas com as áreas, mencionadas acima. Em primeiro lugar decidimos utilizar a linguagem de programação *Python*, versão 2.7. Para todo esse processamento utilizamos as bibliotecas como o *OpenCV* para o processamento das imagens, o *Scikit-Image* para a extração das *features*, o *Scikit-learn* para a classificação e o *Numpy* e *Matplotlib* para auxílio. Dado que nos sentíamos mais familiarizados e mais á vontade para programar. Na Figura 3, podemos observar que, o *drone* irá captar as imagens que serão enviadas para a *ground-station*, onde se situa o operador, que fará todo o processamento das imagens. Detetando onde se localiza o naufrago através de uma indicação gráfica.

# Capítulo 4

## Implementação do Modelo

Nós seres humanos temos a habilidade de o nosso cérebro ter a facilidade de conseguir processar uma imagem e saber distinguir, o que é um objeto do que é o fundo, de uma forma automática, rápida e precisa. Isto não acontece numa máquina, visto que não tem inteligência nem conhecimentos para saber distinguir, de maneira rápida e eficaz. Para resolver esses problemas decidimos dividir as etapas da aplicação em três:

- Fazer a sub-amostragem e dividir em blocos  $M \times N$  de forma a reduzir o custo computacional e definir quais as regiões a serem processadas;
- Aplicar técnicas de segmentação de textura para extrair informação das regiões já definidas e classificá-las;
- Segmentação por cor, para distinguir o objeto.

A sub-amostragem permite reduzir a resolução da imagem, de maneira a que diminua o número de píxeis, como já foi referido anteriormente [16708902312.1]. Isso faz com que o custo computacional reduza significativamente. Para isso utilizamos uma biblioteca de visão computacional associada, à linguagem de programação *Python*, o *OpenCV*. Mesmo assim ainda queríamos reduzir mais, visto que a captura de imagens é feita em tempo real e o custo computacional influencia. Uma vez que a técnica escolhida para a segmentação de textura, é o GLCM, esta técnica não opera para cada *pixel* da imagem, mas sim para um conjunto de *píxeis* com níveis de cinzento. O que nos leva a dividir a imagem em blocos  $M \times N$  e transformar a imagem RGB para GRAY. A dimensão dos blocos vai ser dependente das características que são extraídas dos mesmos, de maneira a diferenciar o que é mar do que não é mar. Depois de recolhidas as características dos blocos utilizando a biblioteca *Scikit-Image*, através do método *greycomatrix*, é feita uma divisão de imagens de treino e imagens de teste. Feita a divisão treinamos o nosso classificador com as características das imagens de treino e o seu respetivo *ground-truth*, e classificamos as imagens de teste, de maneira a verificar se o nosso classificador está a identificar o objeto.

Na Figura 4 é possível visualizar um diagrama que representa todas as etapas feitas até a solução final.

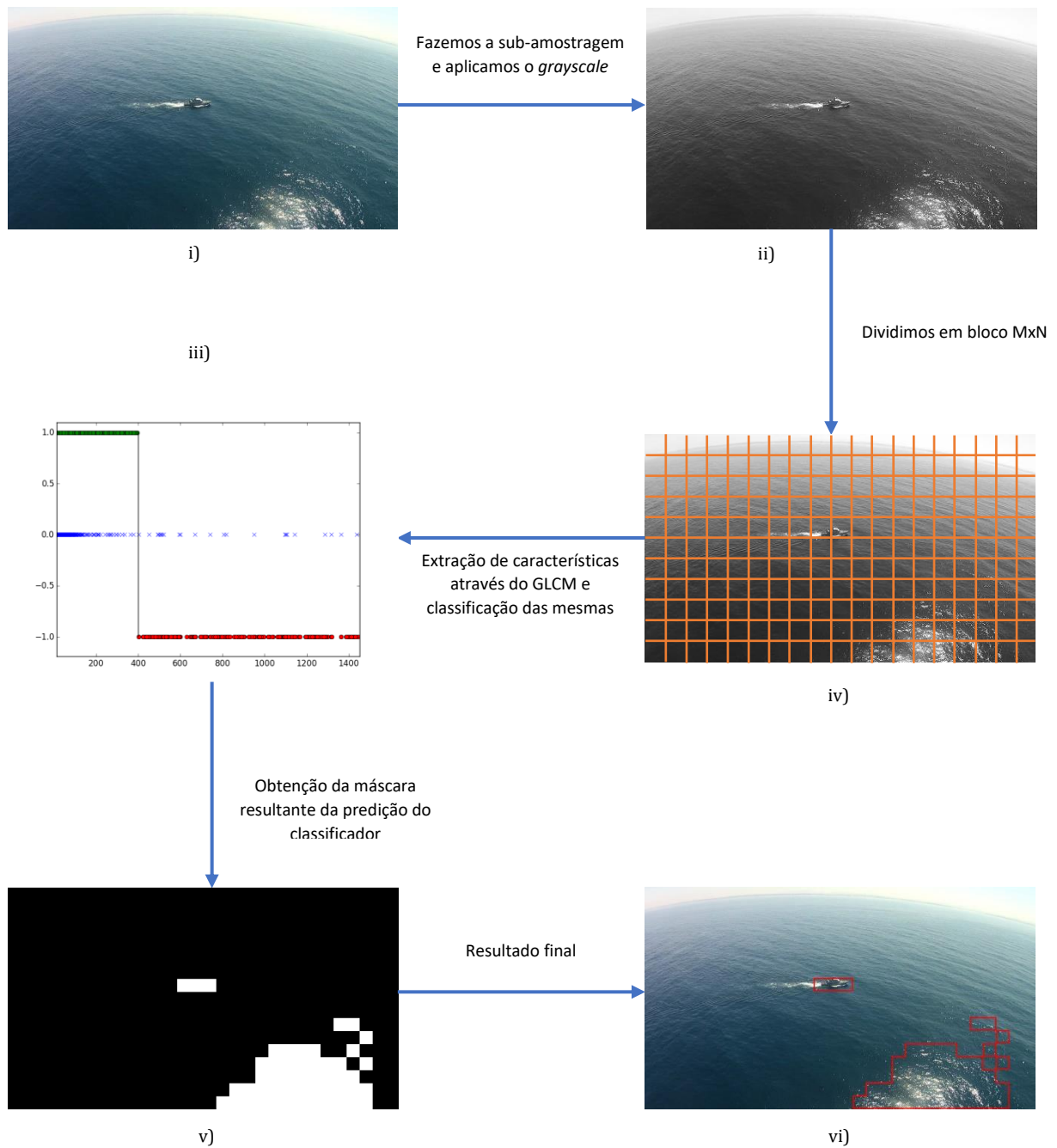


Figura 4 – Diagrama de Blocos da abordagem implementada. i) Imagem Original; ii) Imagem Gray; iii) Imagem dividida em blocos MxN; iv) Classificação das características extraídas do GLCM; v) Máscara resultante da classificação; vi) Imagem com a identificação, do resultado obtido da máscara



## 4.1 Pré-processamento

Com base na tese do [Miguel Griné., 2014], no capítulo de pré-processamento, aplicamos a mesma abordagem, tendo em conta que o objetivo é o mesmo. Utilizamos as técnicas de sub-amostragem e divisão da imagem em blocos. A técnica que nos diferencia é o uso de imagens em tons de cinzento, que é necessário para o *Gray Level Co-Occurrence Matrix*.

### 4.1.1 Plano Gray

Preferimos usar imagens em níveis de cinzento, para reduzir o custo computacional, em vez de usar os planos RGB, de onde se poderia ter Apesar de com o plano RGB obtermos melhores valores, em termos de características, o custo computacional seria elevado. Por esse motivo, preferimos as imagens em níveis de cinzento.

$$RGB \text{ to } Gray : Y = 0.299 . R + 0.587 . G + 0.114 . B \quad (1)$$

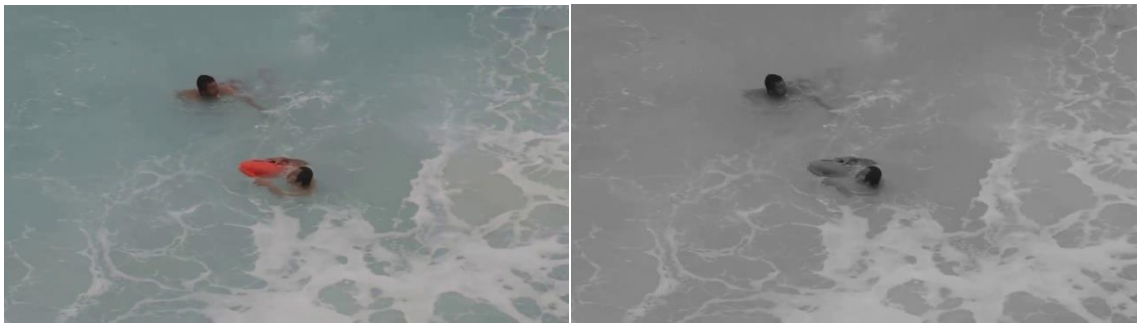


Figura 5 - Imagem original e imagem gray, respetivamente

### 4.1.2 Plano Lab

Surgiu também a hipótese de ser usado o plano 'b' do formato de cor Lab, que corresponde à informação de cromaticidade. Podemos observar, na Figura 6, que as duas pessoas se conseguem diferenciar de tudo o resto. Isso acontece devido à importância que o plano 'b' dá à cor amarela, tornando positivo os pixels com essa cor e negativo com a cor azul.

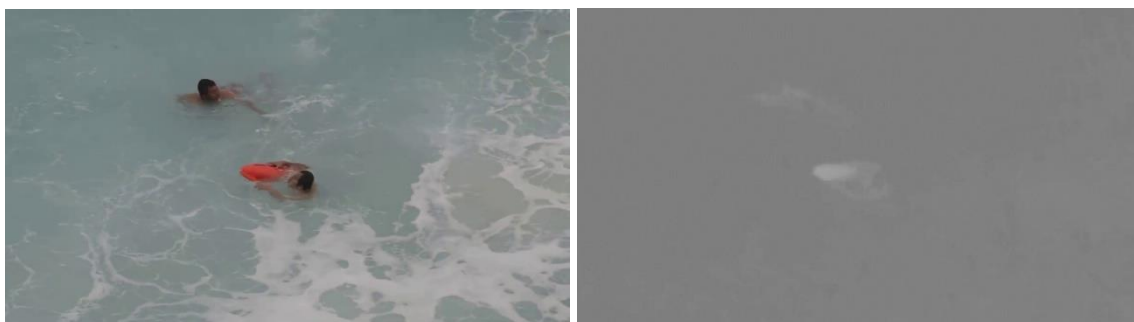


Figura 6-Imagem original e imagem do plano 'b', respectivamente

Segundo o estudo da Tabela 12 confirmou-se pelos valores do contraste apresentados que o plano *Gray* apresenta melhores resultados para diferenciar a pessoa do resto.

Imagem	Características	Plano	Mar Escuro	Mar Claro	Pessoa	Espuma
<b>Frame 3580 (1280x720)</b>	Media	Gray	86.399	157.989	67.953	206.012
		L	98.415	167.098	140.489	210.485
		a	119.426	125.743	126.145	130.626
		b	120.325	123.843	124.490	132.820
	Desvio Padrão	Gray	2.692	1.620	50.826	24.124
		L	2.899	1.558	48.459	22.157
		a	1.614	0.718	5.926	1.1189
		b	1.1349	0.696	8.269	1.1492
	Contraste	Gray	9.313	2.495	561.530	99.367
		L	10.767	2.503	455.809	104.705
		b	1.3507	0.875	7.535	5.840

Tabela 12 – Testes das várias características/features extraídas no GLCM nos vários Planos

---

### 4.1.3 Sub-Amostragem

A sub-amostragem permite diminuir a quantidade de informação a ser processada. Este processo passa por aplicar dois fatores, indicadores de quanto se quer dividir ou multiplicar a resolução da imagem, e uma interpolação *Cubic*, *Nearest*, *Linear* ou *Area*, uma quantificação pixel a pixel consoante a vizinhança.

### 4.1.4 Divisão em blocos

Na segmentação de imagem existem técnicas muito diferentes, que passam por processar pixel a pixel ou com base numa vizinhança e extrair características, como por exemplo, a intensidade média, o contraste, a energia ou a entropia [16708902314.2]. Este varrimento pode ser feito pixel a pixel, mas também com a definição de uma janela/bloco, que irá processar esta mesma informação possivelmente de uma forma mais eficiente e de menor custo computacional. O bloco pode ter várias dimensões, usualmente são blocos simétricos, 16x16 ou 8x8 ou 4x4, consoante o número de pixéis que se quer considerar fazerem parte do mesmo bloco.

### 4.1.5 Teste à sub-amostragem e divisão em blocos

Foram feitas diferentes combinações de pré-processamento para várias imagens da base de dados do vídeo de salvamento aquático. A diferenciação entre imagens com Zoom e sem Zoom teve que ser feita, pois quer-se treinar o classificador com características que variem pouco em relação a todas *frames* do vídeo.

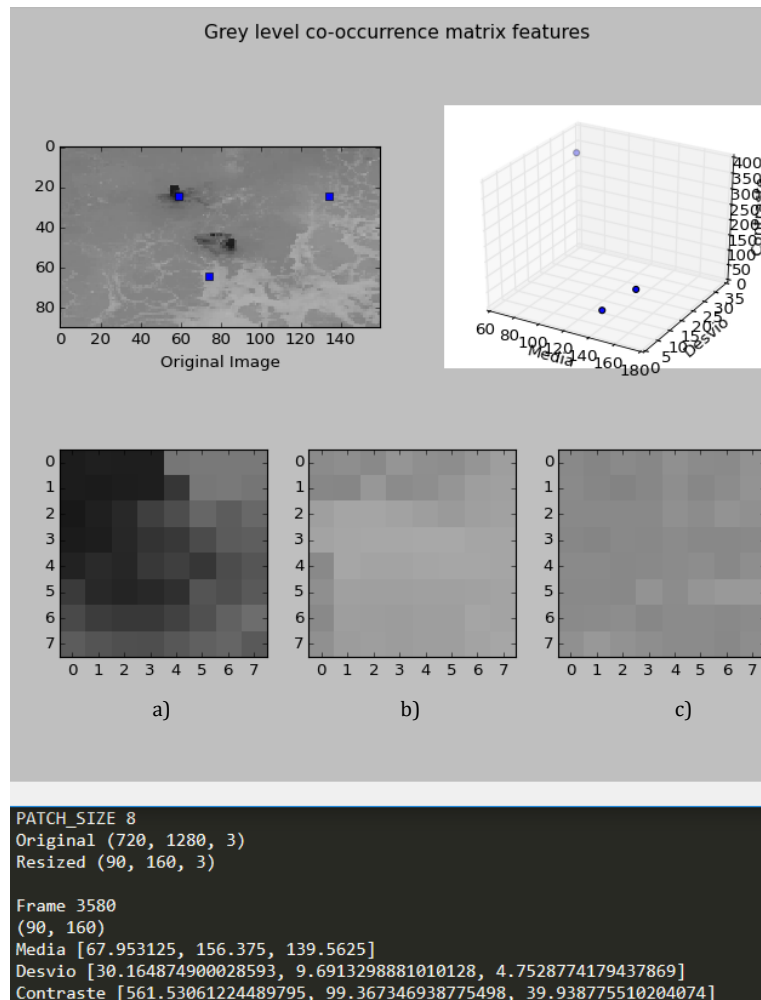


Figura 7 - Frame 3580 em plano Gray dividida por 8, com interpolação Cubic, em blocos de 8x8 na janela do canto superior esquerdo; Representação 3D dos 3 pontos, pessoa, espuma e mar, no canto superior direito com as features Media, Desvio e Contraste (a), (b) e (c).

Esta é a combinação dos métodos selecionada, para imagens com Zoom, divisão da imagem por 8 e por blocos de 8x8. Como se pode ver na Figura 8, todos os blocos que dividem por 2 e por 4 mostram ainda muitos níveis de cinzento, impossibilitando adquirir melhores valores de contraste.

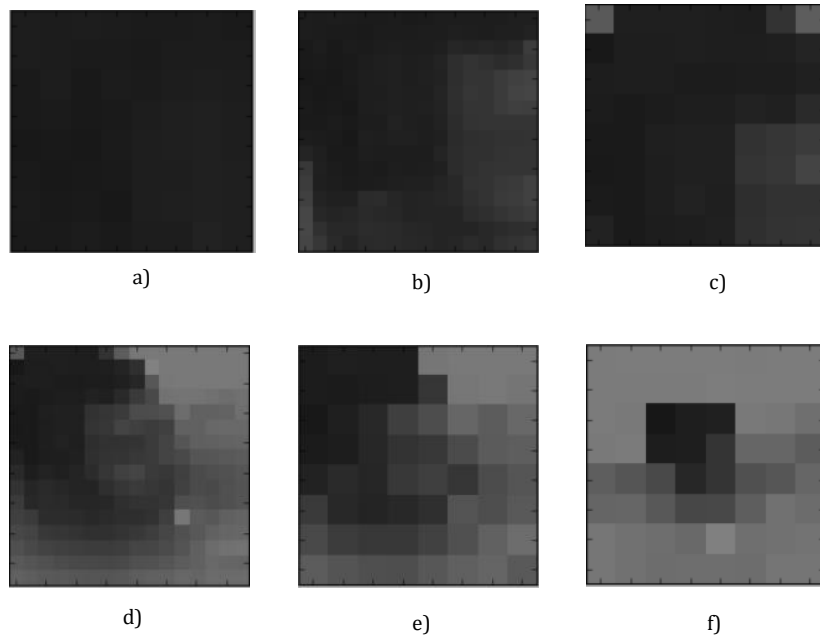


Figura 8 - bloco 8x8 e imagem/2 (a); bloco 16x16 e imagem/2 (b); bloco 8x8 e imagem/4 (c); bloco 16x16 e imagem/4 (d); blocos 8x8 e imagem/4 (e); bloco 8x8 e imagem/8 (f).

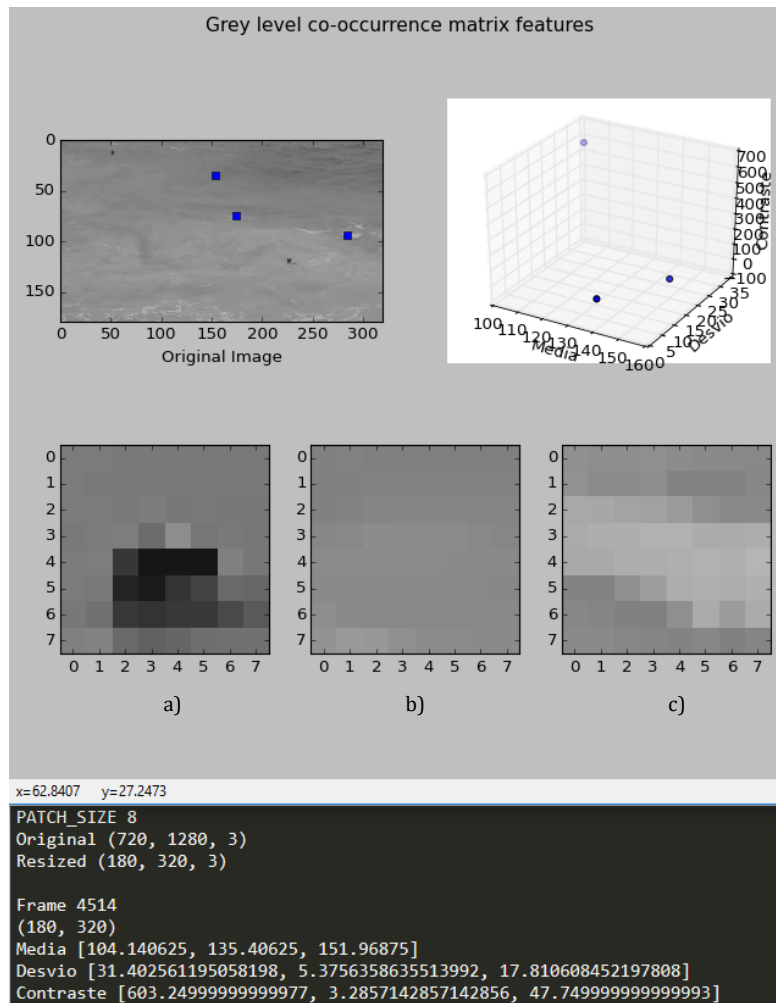


Figura 9 - Frame 4514 em plano Gray dividida por 8, com interpolação Cubic, em blocos de 8x8 na janela do canto superior esquerdo; Representação 3d dos 3 pontos, pessoa, espuma e mar, no canto superior direito com as features Media, Desvio e Contraste (a), (b) e (c)

Na Frame 4514, que considerámos não ter Zoom, a melhor divisão de imagem é por 4 e de blocos por 8, mantendo-se aqui um padrão no tamanho do bloco, tamanho esse que vai também ser influenciado pela dimensão da imagem, neste caso será 9x8.

## 4.2 Extração de Características

Como foi referido na secção 16708902313.1 para a extração de características utilizamos o GLCM. O GLCM é uma matriz que indica o número ocorrências que o nível de cinzento, representado por uma etiqueta, ocorre a uma distância  $d$  e a um ângulo  $\alpha$  como podemos visualizar na Tabela 13.

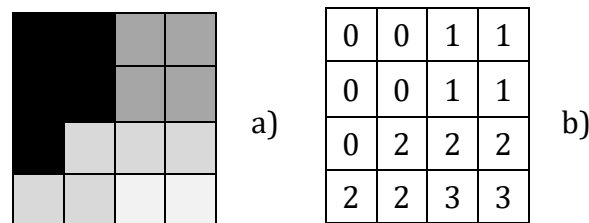


Figura 10 - Blocos de níveis de cinzento e bloco de etiquetas de acordo com os níveis de cinzentos, respetivamente

Tendo a matriz de etiquetas, Figura 10(b), é calculada a matriz de co-ocorrência. Definindo qual a distância e qual o ângulo obtém-se o número de relações espaciais definidas na Figura 11.

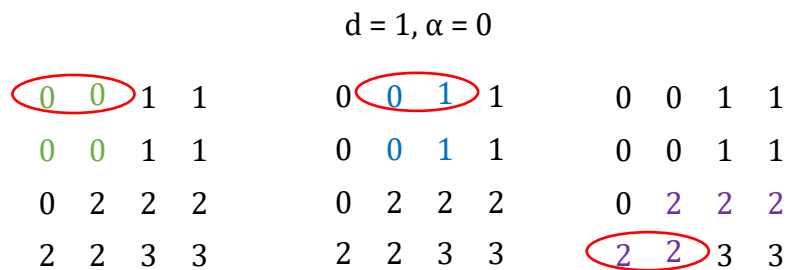


Figura 11 – Exemplo de grupo de pixéis, de níveis de cinzento, com maior número de ocorrências na imagem

Etiquetas	0	1	2	3
0	2	2	1	0
1	0	2	0	0
2	0	0	3	1
3	0	0	0	1

Tabela 13 - Matriz de coocorrência

Obtida a matriz de coocorrência, podemos então calcular as propriedades de textura, como por exemplo:

$$contrast = \sum_{i,j=0}^{levels-1} P_{i,j} (i - j)^2 \quad (2.1)$$

$$dissimilarity = \sum_{i,j=0}^{levels-1} P_{i,j} |i - j| \quad (2.2)$$

$$homogeneity = \sum_{i,j=0}^{levels-1} \frac{P_{i,j}}{1 + (i - j)^2} \quad (2.3)$$

#### 4.2.1 Decisão da distância e ângulo a utilizar

Tendo em conta a dimensão do bloco MxN, a distância máxima que poderia ser percorrida seria o número M ou N, dependendo de qual o maior, por exemplo 8. O ângulo pode variar entre 0, 45, 90 e 135 graus.

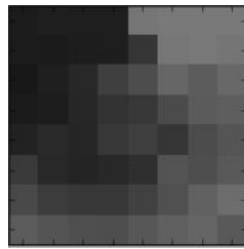


Figura 12 - Bloco de MxN

Uma variação da distância implicaria um maior custo computacional para o nosso processamento, e por isso escolhemos fazer a comparação de níveis de cinzento a uma distância de 1.

Segundo a Tabela 14 que retira os valores do bloco da Figura 12, representativo de uma pessoa, concluímos que o ângulo a utilizar será o 135.

Ângulos	0	45	90	135
Contraste	349.64	345.43	306.27	561.53

Tabela 14 - Ângulos e respetivos contrastes da Figura 12



## 4.3 Classificação

Extraídas as características, é necessário saber distinguir do que é mar e não mar. Para isso foi necessário o uso de técnicas de Aprendizagem Automática. Utilizamos a biblioteca *Scikit-learn*. Esta biblioteca fornece vários métodos de classificação e regressão, entre outras funções para o tratamento da informação.

SVM, a escolha para o nosso classificador, é um algoritmo de aprendizagem automática supervisionado. O seu objetivo é encontrar um hiperplano que separe as duas classes e apresenta a maior distância ao padrão de qualquer classe mais próxima, de maneira a minimizar o erro. Na Figura 13, podemos observar o hiperplano  $L_1$  separa em diferentes classes o *cluster* de cor azul, o que faz com que haja, erro na classificação. O hiperplano  $L_2$  já consegue separar corretamente os padrões das classes de cor azul e vermelha, contudo é o hiperplano  $L_3$  que tem as maiores distâncias aos padrões mais próximos das diferentes classes, chamados de vetores de suporte.

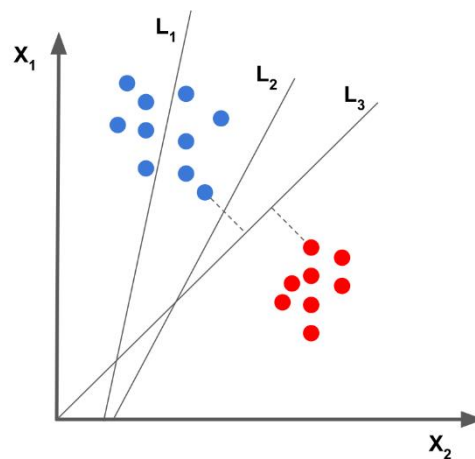


Figura 13 - Classificador SVM

### 4.3.1 Classificação SEAGULL

Utilizando a base de dados da SEAGULL, e aplicando o processo já referido na Figura 4, consegue-se o resultado da imagem vi), concluindo que o contraste de facto é uma boa característica para salientar o mar. O próximo passo seria retirar toda a espuma e reflexão do sol para ficar unicamente com a pessoa.

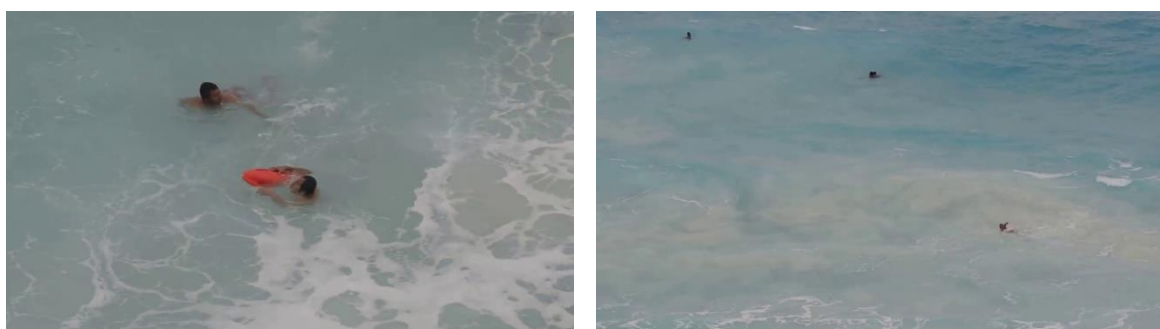
Passamos então à segmentação por cor, técnica referida na secção 4.1.1 Plano Gray, que foi a solução encontrada para o tipo de características que achámos relevantes para diferenciar o que é pessoa e todo o ruído no mar. Conforme a tabela seguinte apresentada, a média e o desvio vieram representar o que seria uma boa forma de juntar este ruído ao cluster de mar, para manter uma classificação binária e assim obter só a pessoa.

	Pessoa	Sol	Espuma
<b>Média</b>	93.234	164.375	142.347
<b>Desvio Padrão</b>	46.107	41.692	56.459
<b>Contraste</b>	1231.079	2210.758	490.745

*Tabela 15 - Novas características diferenciadoras*

### 4.3.2 Classificação Final

Para treinar o classificador, com as novas características, decidimos utilizar uma base de dados já parecida com o tipo de missão em estudo. Começamos por separar as imagens de treino das imagens de teste. As imagens de treino foram escolhidas aleatoriamente, do conjunto de imagens com Zoom e do conjunto sem Zoom:



*Figura 14 - Imagens de Treino - Frame 3580 e Frame 4512, respetivamente*

De seguida calculamos os *groundtruth* das imagens de treino, em dividido em 2 classes (Mar, Espuma, Sol) e Pessoa. Aplicamos depois a sub-amostragem e a divisão por blocos, às imagens mencionadas na Figura 14, para calcularmos as características. A *frame* 3580 obteve 200 blocos e a *frame* 4514 obteve 800 blocos. Em cada bloco foram extraídas as características, contraste, média e desvio. Concluimos que, a *frame* 3580 tem 200 amostras, com 3 dimensões e a *frame* 4514 tem 800 amostras, com 3 dimensões. Antes de treinarmos o classificador, dividimos os *groundtruth* em 2 classes, definindo o que é pessoa e não pessoa. Dado que a distância entre os *clusters* é significativa concluimos que o *kernel* a utilizar no classificador é o linear, visto que se trata de uma classificação binária.

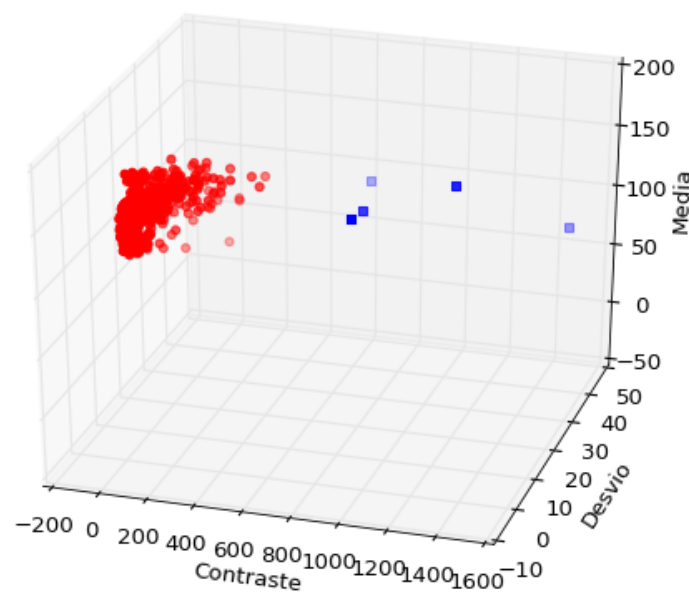


Figura 15 - Features extraídas

## 4.4 Métodos Combinados para melhor identificação

Segundo os resultados que irão ser apresentados no capítulo 5, foram desenvolvidas duas técnicas para melhorar a identificação da pessoa e porventura melhorar a classificação.

### 4.4.1 Técnica de segmentação por cor adaptada

De maneira a reduzir possíveis erros do classificador e otimizar o algoritmo, é feito um *scanning* aos blocos classificados como verdadeiros onde se percorre todos os pixels. Caso o pixel tenha um valor de intensidade abaixo de um *threshold* que consideramos mais perto de preto, esse bloco é considerado pessoa.

### 4.4.2 *Tracking*

O *tracking* surgiu para criar alguma coerência na sequência das *frames*, em que por vezes a pessoa pode ser perdida, por todos os motivos já mencionados do pré-processamento. A solução passa por criar três listas, blocos desaparecidos, blocos considerados pessoa pelo SVM, e novas localizações para os blocos desaparecidos. Ao verificar se os novos blocos que foram classificados se mantêm em relação aos antigos, aplico uma nova localização aos que já não existem. Aos blocos desaparecidos que sejam iguais aos novos classificados, são retirados da lista de desaparecidos. E finalmente todas as novas localizações e blocos atuais são ajustados.

Este ajuste vem compensar a deficiência do que pode ser, dividir a imagem em blocos, pois se o alvo a identificar não tiver o melhor enquadramento no bloco, pode não obter os melhores valores para ser classificado como pessoa. Para um melhor enquadramento do alvo, efetua-se um *scanning* à volta do bloco já identificado, com uma janela de procura de  $M \times N$ , e o bloco que tiver melhor contraste do que o bloco inicial será apresentado.

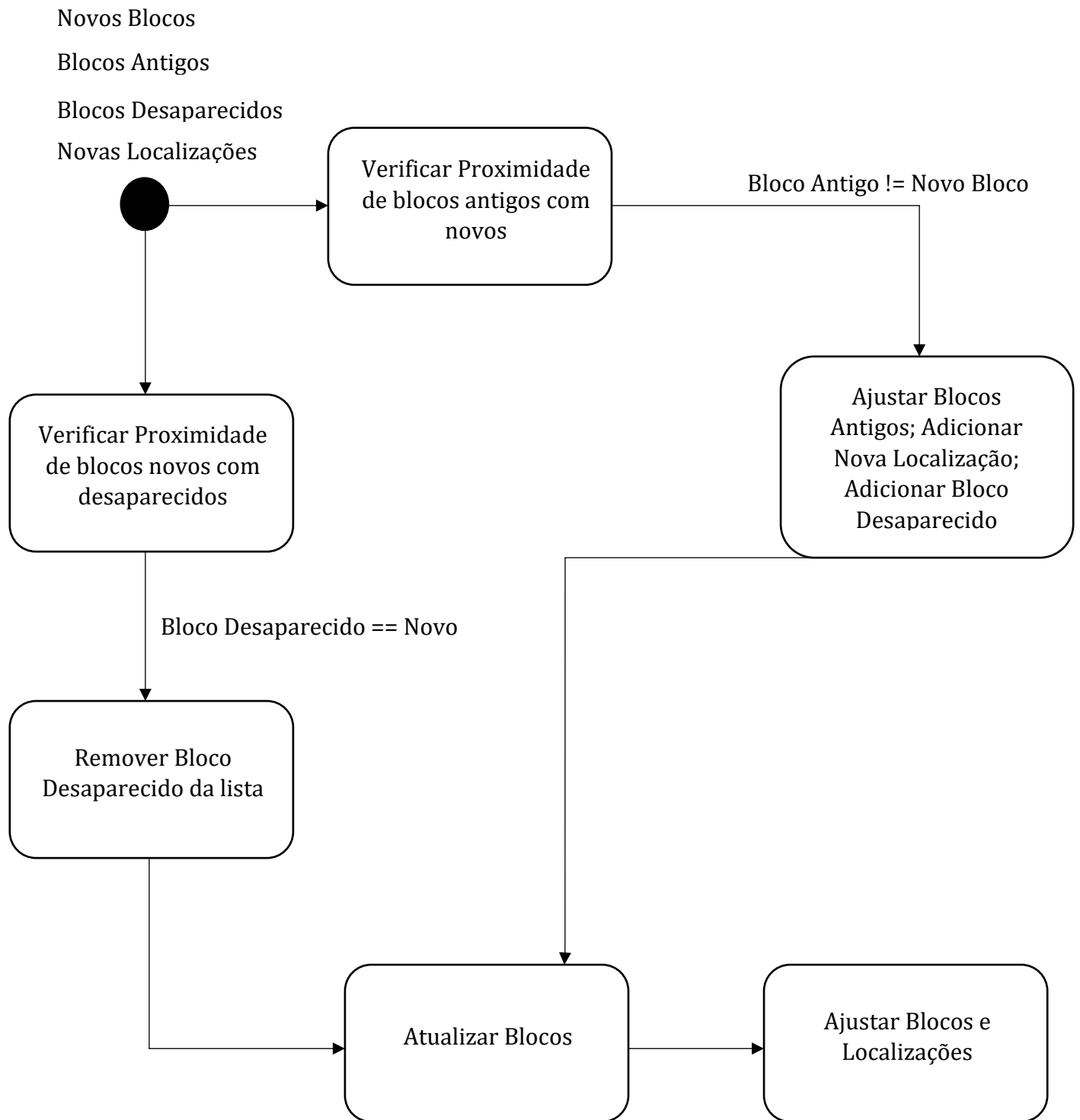
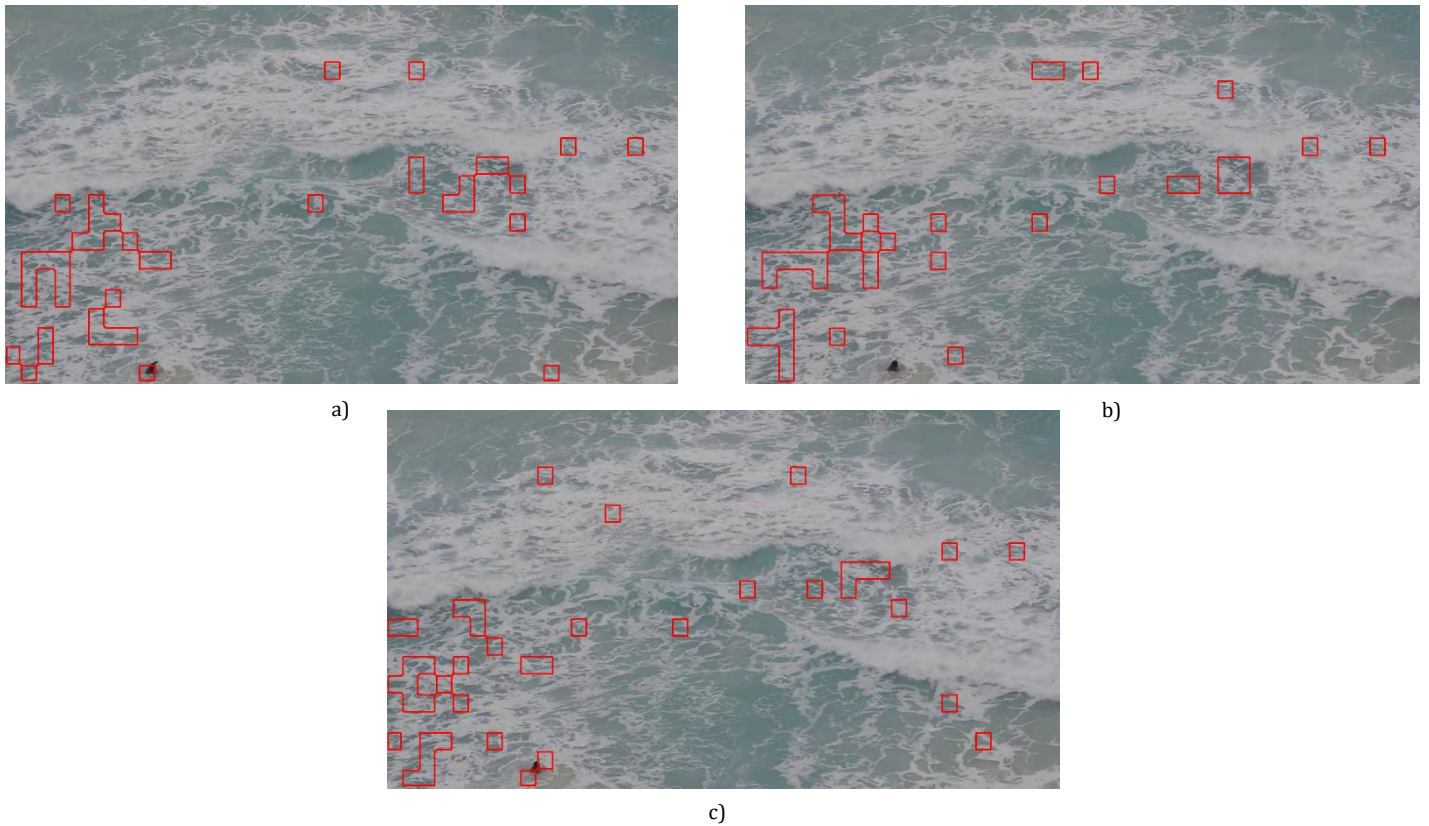
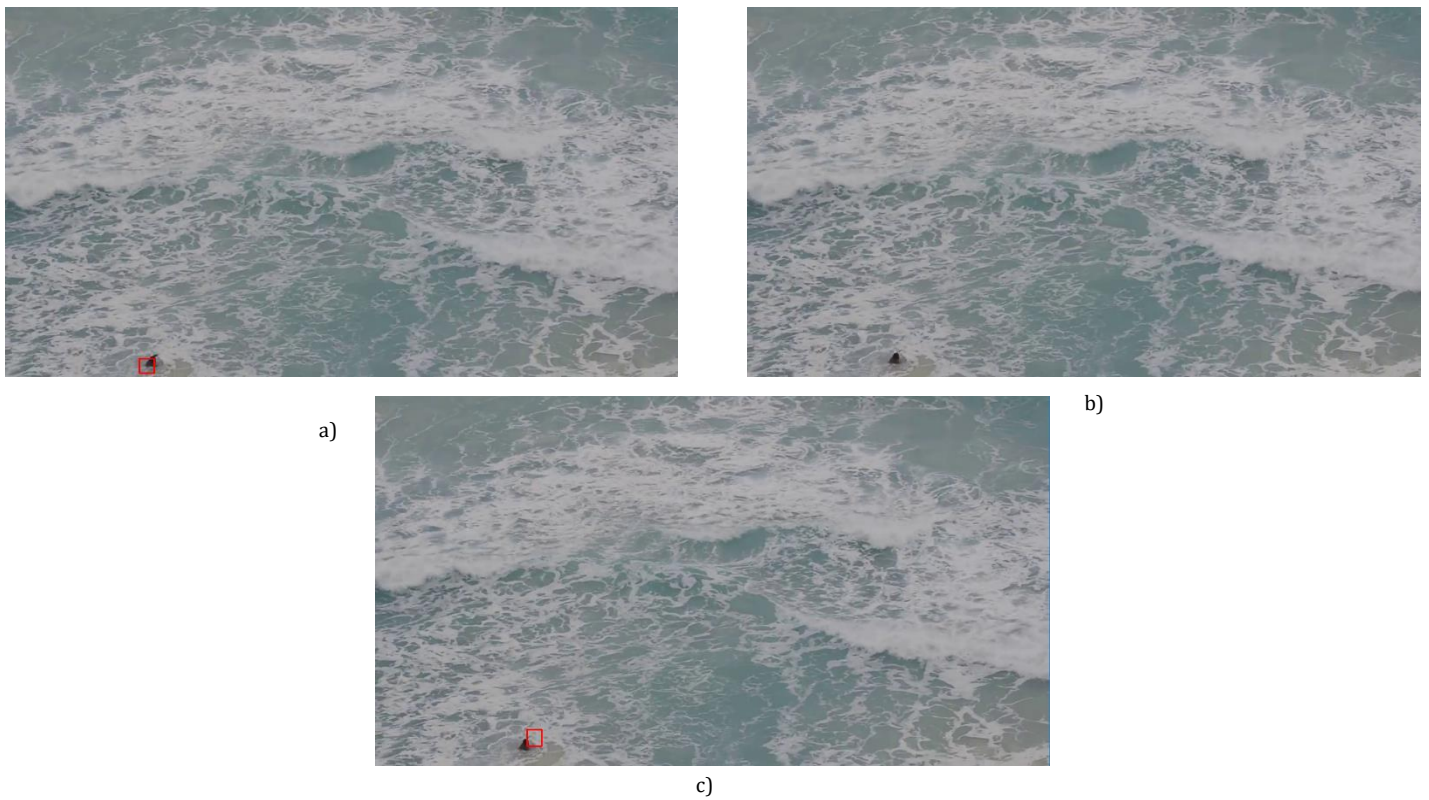
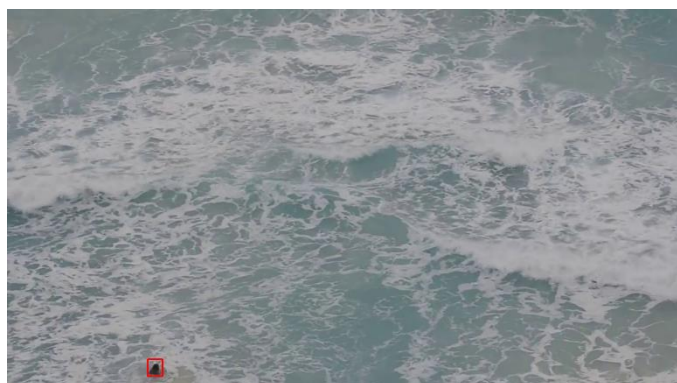


Figura 16 - Diagrama de estados do Tracking

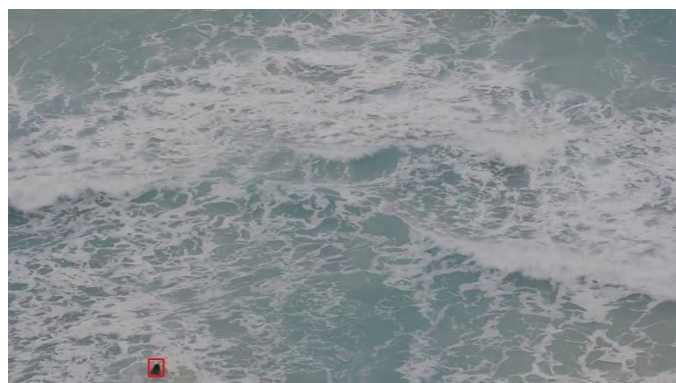
## 4.4.3 Processamento de Frames com os métodos combinados

*Figura 17 - Frames mal classificadas**Figura 18 - Sequência das mesmas frames, que a Figura 16, mas com a técnica de segmentação de cor adaptada*

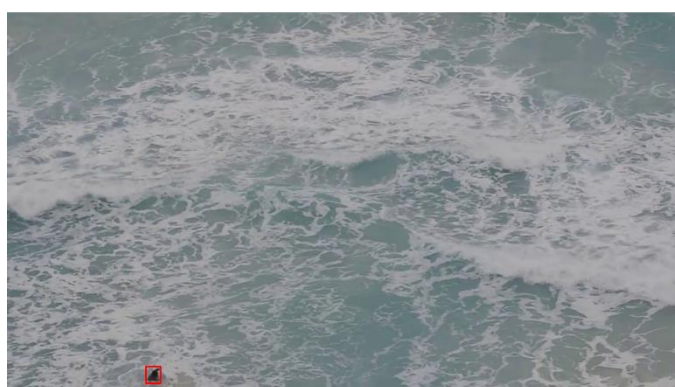




a)



b)



c)

*Figura 19 - Sequência de frames com o tracking, onde se vê que na imagem b) a pessoa já é identificada*





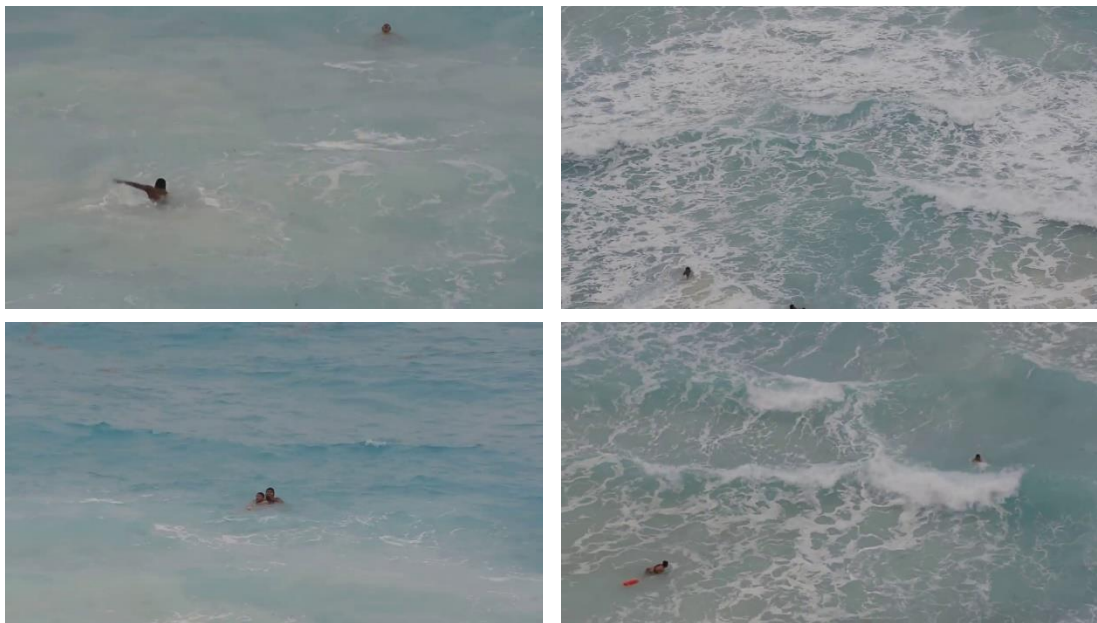
# Capítulo 5

## Validação e Testes

Este capítulo apresenta os testes e resultados obtidos a partir do modelo proposto e dos métodos descritos nos capítulos 3 e 4. Os métodos de avaliação são também aqui explicados. E por fim, os resultados conseguidos para cada sequência de imagens.

### 5.1 Base de dados

O primeiro vídeo a ser utilizado para testes, é o principal pois foi com este que todo o modelo foi treinado, é um vídeo de salvamento aquático, do qual podemos retirar várias sequências de imagens, com diferentes estados do mar e diferentes zoom. O segundo vídeo retiramos uma sequência de imagens para testar, tem um surfista e as características do mar mostram-se diferentes do primeiro vídeo.



*Figura 20- 4 imagens de 4 sequências de teste do vídeo de salvamento*



Figura 21- Imagem da sequência de imagens do vídeo do surfista

Das imagens apresentadas, irão também ser testadas as imagens consecutivas, mas de 3 em 3, ou seja, para representar uma sequência de imagens de um vídeo, utilizamos a imagem 1895, a 1898, a 1901 e a 1904, de forma a haver mais variação intrasequencial.

## 5.2 Métodos de Avaliação

Em reconhecimento de padrões e classificação binária principalmente, é utilizada a *precision*, que é a fração do que é relevante e foi identificado como positivo (TP), sobre tudo o que foi identificado (TP+FP), e o *recall*, que é a fração do que é relevante e foi identificado (TP), sobre tudo o que é relevante e foi identificado mais o que não foi identificado (TP+FN). Utiliza-se também a métrica *accuracy* que representa a medida de proximidade aos valores reais, e o *f-score*, ambos tomam em consideração o *precision* e o *recall*. As fórmulas podem-se representar por:

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$fscore = \frac{precision * recall}{precision + recall}$$

## 5.3 Testes Efetuados

Como já foi previamente referenciado no capítulo 4, houve uma evolução do querer salientar as regiões que eram mar das que não eram, ao utilizar o contraste para uma classificação final do modelo utilizando mais duas características que conseguissem classificar a pessoa. Mesmo com o classificador finalmente treinado e orientado para identificar o que é pessoa, espera-se que os nossos FPs sejam espuma e reflexo do sol, mas que o nosso método adaptado para segmentação por cor, resolva a maioria ou a totalidade. Espera-se também FNs, causados pelo enquadramento do bloco e da pessoa, e para colmatar a desvantagem de dividir a imagem por blocos, faz-se o *tracking* dos possíveis FNs, para reduzir o mais possível, porque com mais detalhe, chegamos à conclusão que todos os FNs serão blocos confundidos com espuma e que nunca apareçam.

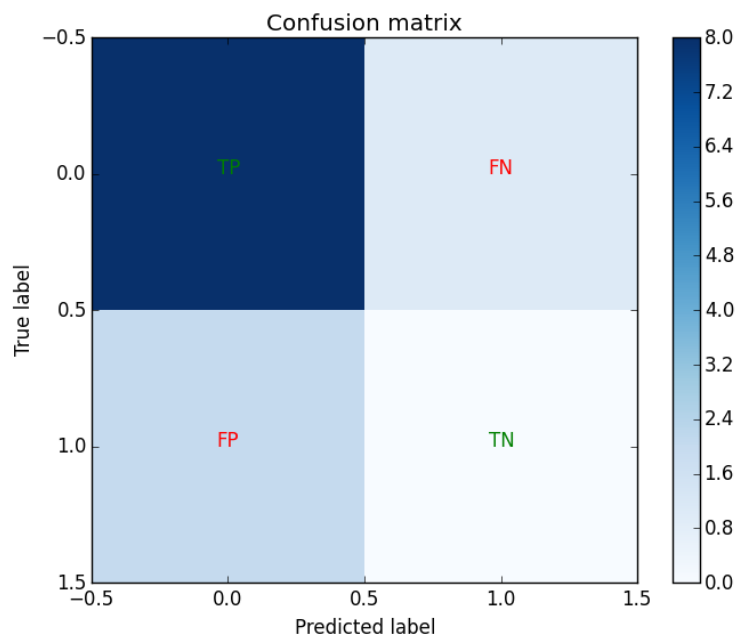


Figura 22 - Matriz de confusão padrão

Apresentamos de seguida todas as matrizes de confusão, *precision*, *recall*, *accuracy* e *f-score*.

- Sequência de imagens a partir da imagem 117

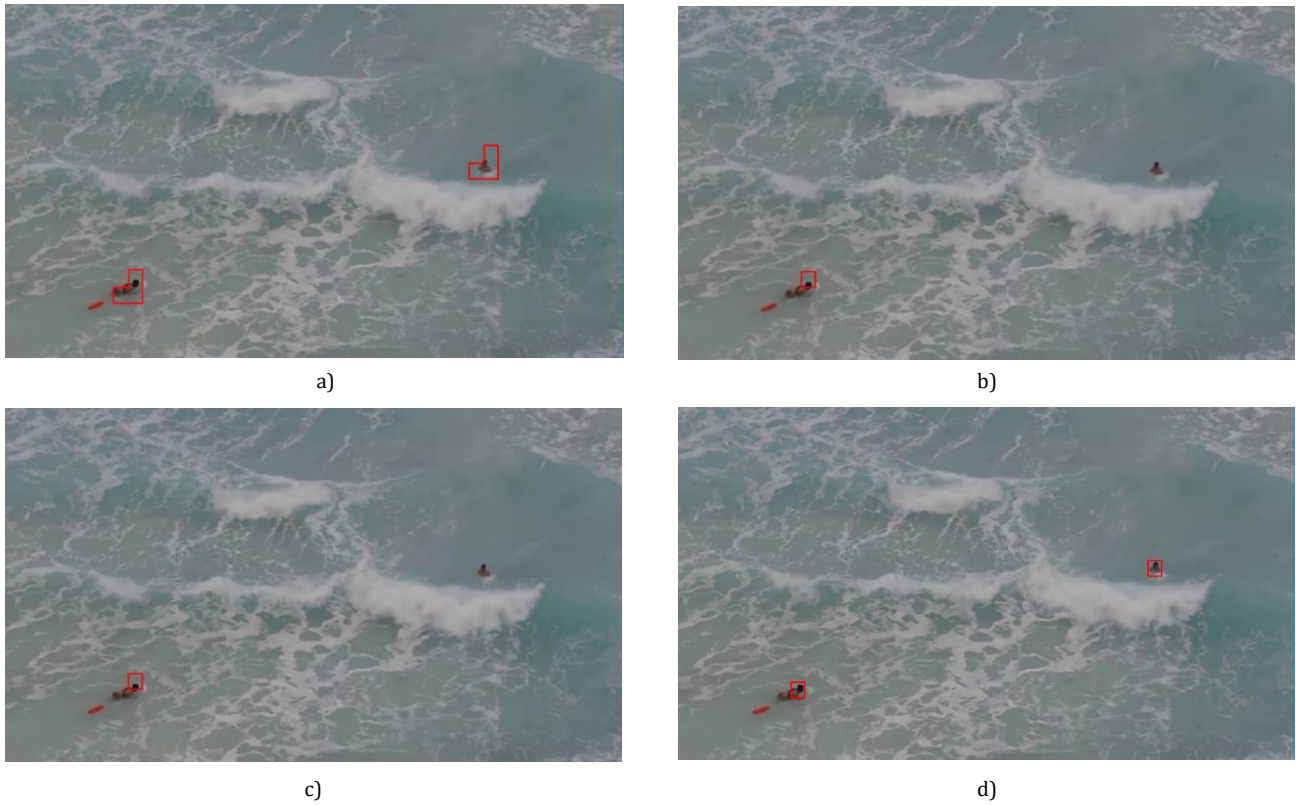


Figura 23 - Ground-thruth (a), Classificador (b), Ajuste (c), Tracking (d)

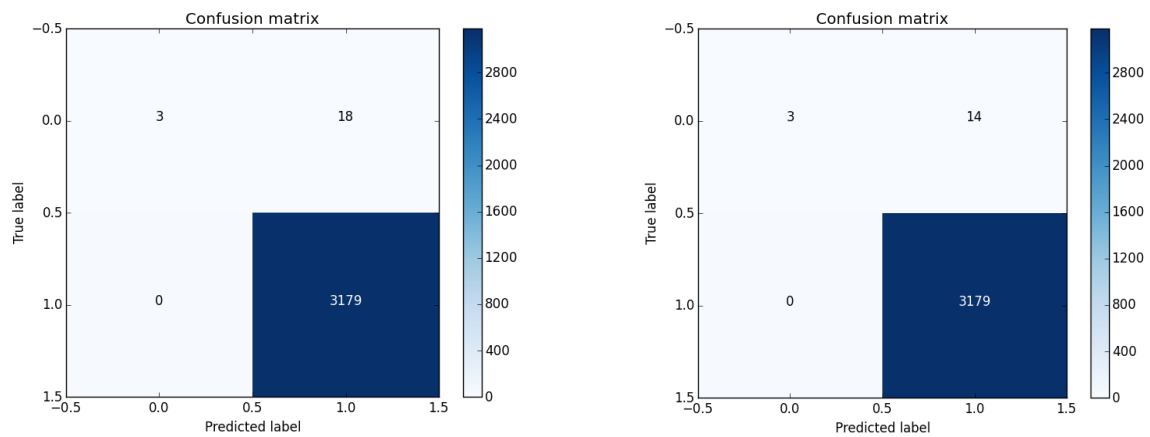


Figura 24 - Matriz de co-ocorrência das imagens (b) e (d)

	Classificador	Técnica Adap.	Tracking
<i>Recall</i>	0.143	0.176	0.176
<i>Precision</i>	1.0	1.0	1.0
<i>Accuracy</i>	99%	99%	99%
<i>F-score</i>	0.25	0.3	0.3

- Sequência de imagens a partir da imagem 1892

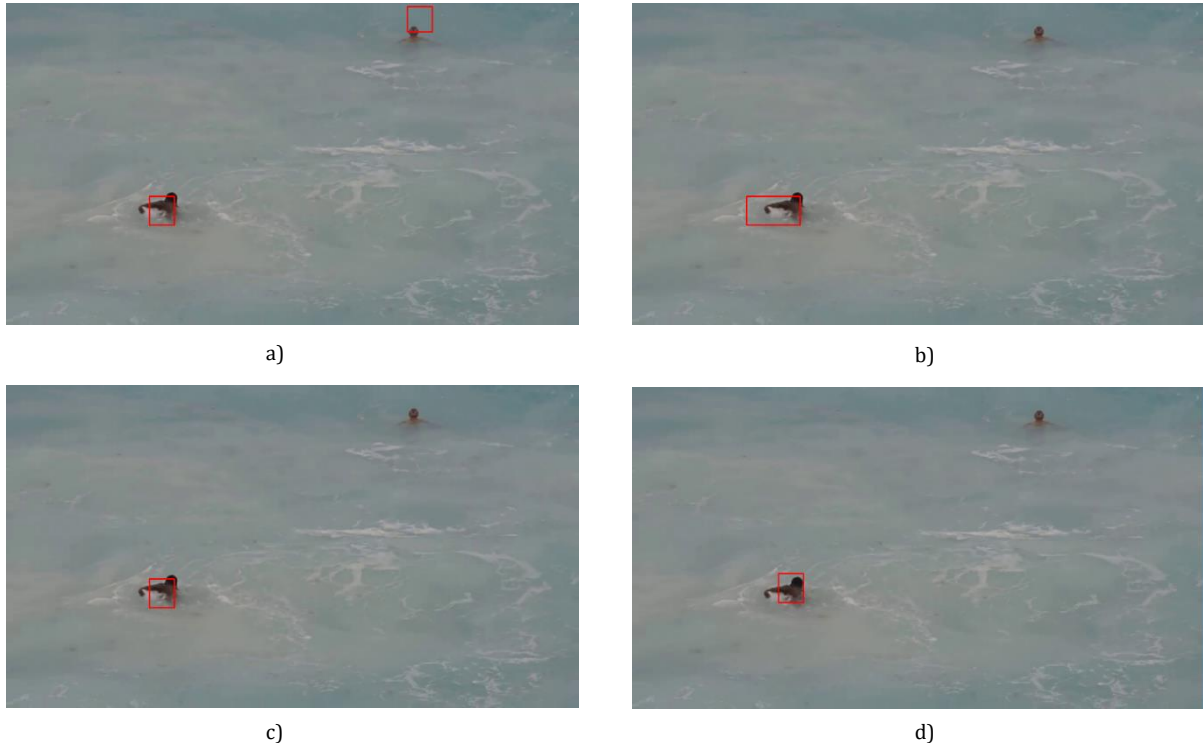


Figura 26 - Ground-thruth (a), Classificador (b), c/Ajuste (c), c/ Tracking (d)

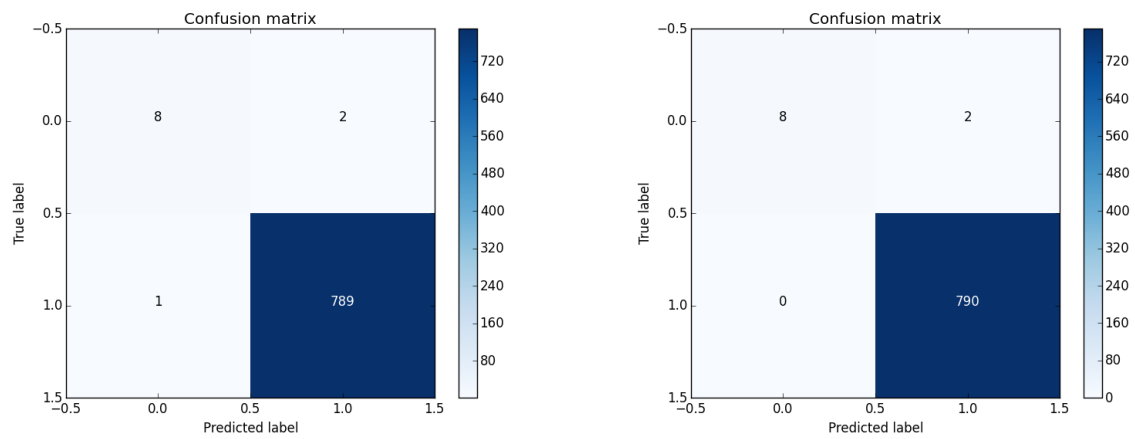


Figura 25 - Matriz de co-ocorrência das imagens (b) e (d)

	Classificador	Técnica Adap.	Tracking
Recall	0.8	0.8	0.8
Precision	0.89	1.0	1.0
Accuracy	99%	99%	99%
F-score	0.84	0.89	0.89

- Sequência de imagens a partir da imagem 2494

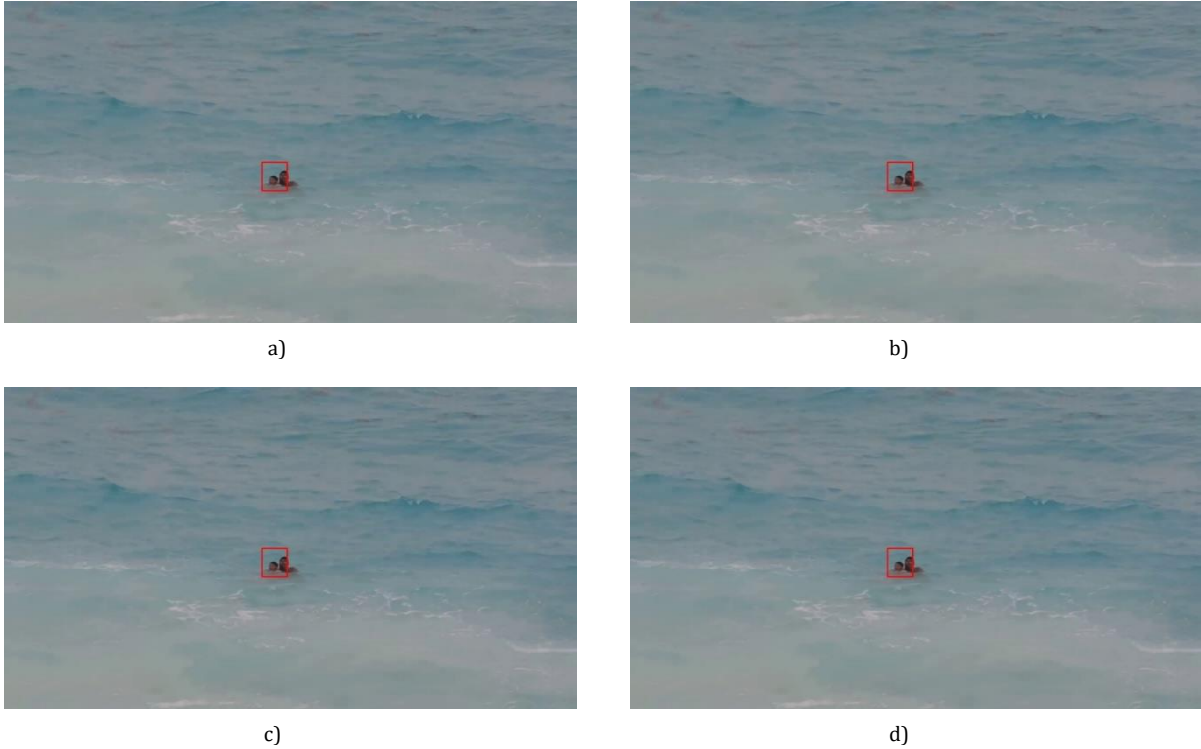


Figura 27 - Ground-thruth (a), Classificador (b), c/Ajuste (c), c/Tracking (d)

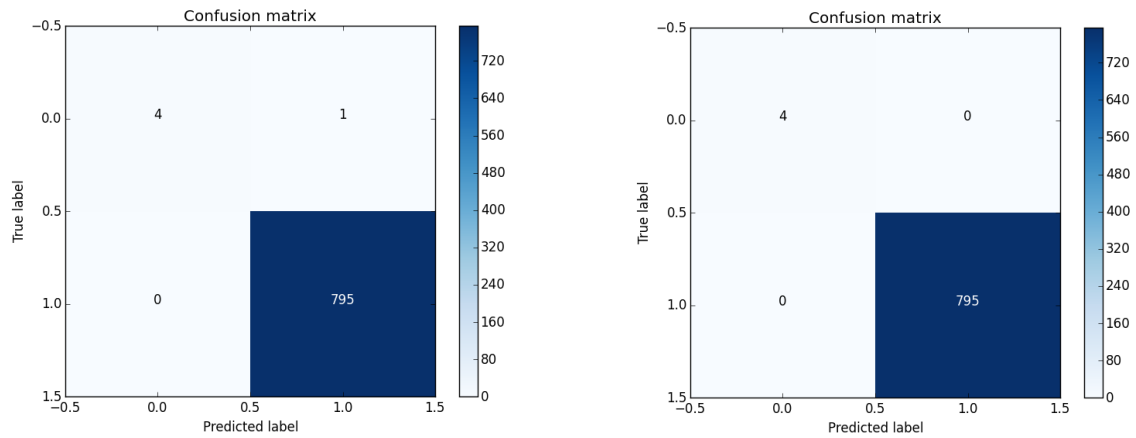


Figura 28 - Matriz de co-ocorrência das imagens (b) e (d)

	Classificador	Técnica Adap.	Tracking
<i>Recall</i>	0.8	1.0	1.0
<i>Precision</i>	1.0	1.0	1.0
<i>Accuracy</i>	99%	100%	100%
<i>F-score</i>	0.89	1.0	1.0



- Sequência de imagens a partir da imagem 673

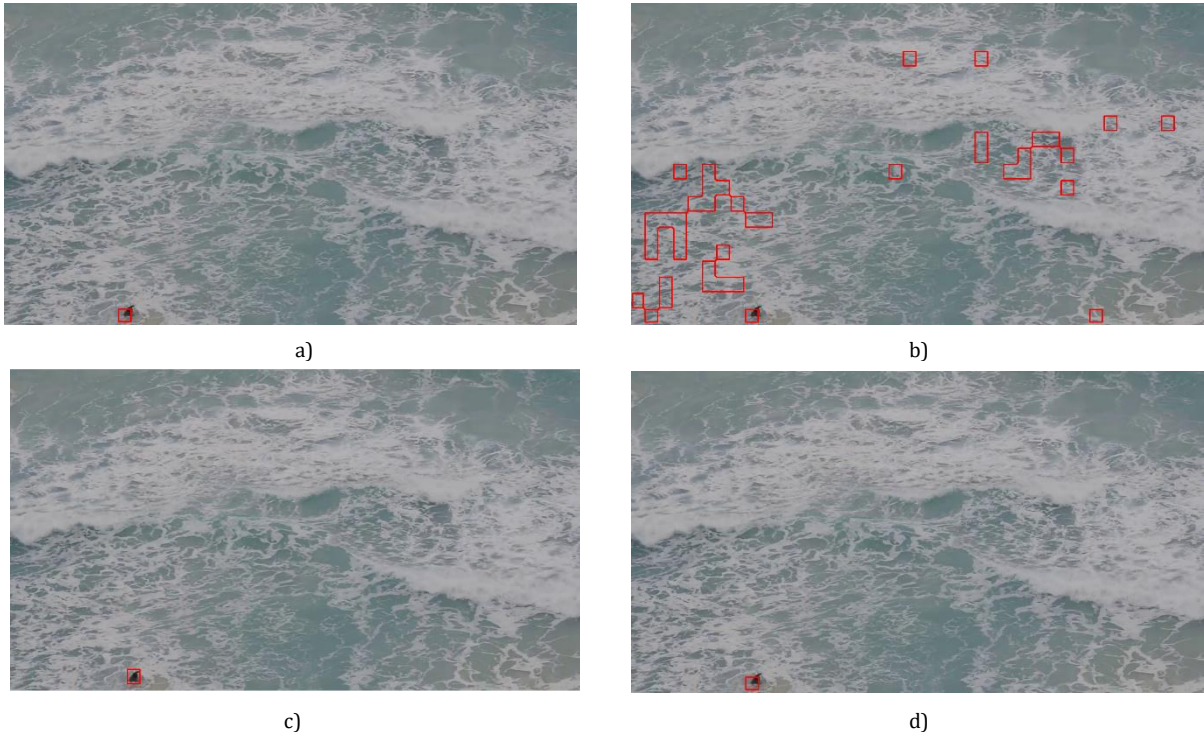


Figura 30 - Ground-thruth (a), Classificador (b), c/Ajuste (c), c/Tracking (d)

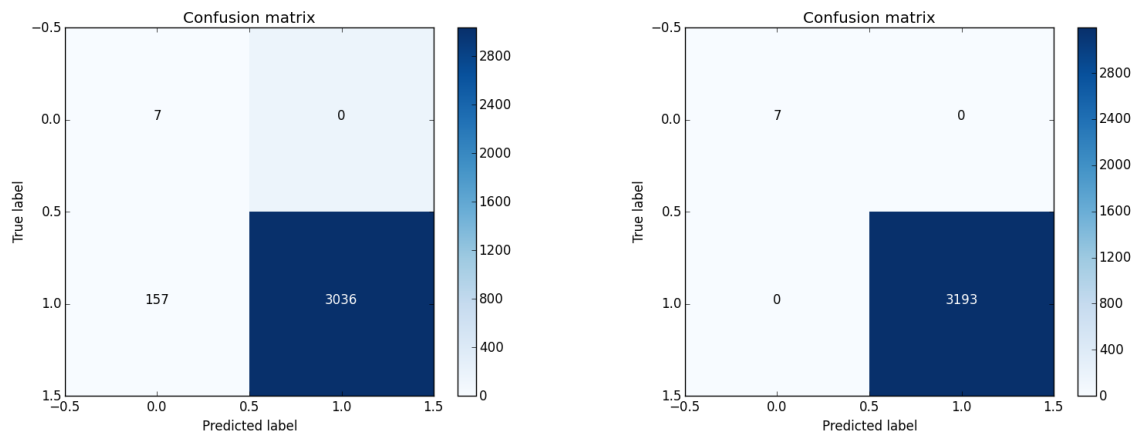


Figura 29 - Matriz de co-ocorrência das imagens (b) e (d)

	Classificador	Técnica Adap.	Tracking
Recall	1.0	1.0	1.0
Precision	0.04	1.0	1.0
Accuracy	95%	100%	100%
F-score	0.08	1.0	1.0

Destes testes e resultados, retira-se que o classificador tem um *recall* de 0,685, uma *precision* de 0,732, um *accuracy* de 98% e um f-score de 0,508. De um modo geral as técnicas combinadas melhoram estas métricas.

Como limitações nota-se que por exemplo na sequência da 1892, o *tracking* não consegue minimizar os FN, porque nas imagens seguintes, o classificador nunca consegue classificar a segunda vítima.

A seguinte Figura mostra como o nosso classificador classifica a sequência de imagens a partir da imagem do surfista 614:



Figura 31- Imagem 614 classificada com o classificador treinado com a base de dados do video do salvamento



## Capítulo 6

### Conclusões e Trabalho Futuro

Foi projetado conseguirmos um sistema semiautónomo para deteção de náufragos em zona costeira, utilizando um *drone*. Com o *know how* adquirido durante o trabalho, foram-nos apresentadas várias limitações para o resultado final que gostaríamos de ter, a nível de implementação do sistema. Primeiro, qualquer tipo de *drone* para realizar filmagens em zonas protegidas como praias/zona costeira, precisa de uma autorização da AAN e ICNF, para voo nestas zonas com mais de 12 pessoas, é preciso uma autorização da ANAC, como mencionado nas regras de voo disponibilizadas no site [VoanaBoa, 2018]. Segundo, o DJI Inspire 1 adquirido pela empresa MicroSegur, não é *OpenSDK*, o que não nos permitiu integrar no computador a bordo do mesmo, o nosso algoritmo, que tornaria este sistema acima de tudo realista. Terceiro, tivemos que simular muitas técnicas de processamento, que seriam efetuadas a bordo do *drone*, como retirar informação da câmara para saber quando é feito Zoom, e consequentemente fazer o pré-processamento baseado neste feedback. Quarto, o envio de imagens para o operador é agora feito após a captação das mesmas, pois o processamento é feito na GCS e não no próprio *drone*.

Para resolver estas limitações, tivemos de reunir vídeos e imagens de outras bases de dados. Processamos as mesmas imagens diretamente no computador que é a nossa GCS, e todos os métodos dependentes de saber se a imagem tem Zoom aplicado ou não, fazemos manualmente.

Por ser uma classificação supervisionada foi necessário fazer *groundtruth* só algumas imagens do vídeo, isto pelo pouco tempo que se tinha para poder classificar todas as imagens e por cada uma ter entre 200 a 800 blocos para classificar manualmente. As imagens selecionadas representam várias sequências de imagens aleatórias, para simular o que seria uma separação das mesmas em conjuntos de treino e teste, caso tivéssemos todas as imagens com o *groundtruth* feito. Ao não ter todos os *groundtruth*, não conseguimos mostrar no vídeo, a pessoa com contorno verde, mostrando que é bem classificado, estando todos os blocos identificados, a vermelho.

Sabendo a desvantagem da divisão por blocos, a razão para os nossos Falsos Positivos e Falsos Negativos é os blocos que não conseguem apanhar grande parte da pessoa e assumem um contraste parecido com mar ou espuma, e a característica que seria diferenciadora, a média, aproxima-se do mar caso tenha valores abaixo de 120, e da espuma caso tenha valores acima de 150.

Para trabalho futuro queremos reduzir o tempo na extração de características, pois o GLCM demora 3s por imagem sem Zoom e 1s por imagem com Zoom. Tempo esse que também podia ser reduzido se tivéssemos um melhor processador, mas de notar que na aplicação real, seria com o processador a bordo do *drone*, com menor custo computacional, ou seja, é prioritário substituir o GLCM.

A segmentação da imagem não seria por blocos, mas sim pixel a pixel, para não obter tantos falsos positivos e falsos negativos já justificados anteriormente.

O reconhecimento do comportamento da pessoa, é uma área de possível estudo e melhoria do projeto, em que os FN serão pessoas que não sejam vítimas, ou seja, não aparentam necessitar de ajuda.

Finalizando com o objetivo principal de continuar com este projeto, é a possibilidade de criar um sistema de patrulha totalmente autónomo e inteligente, constituído por mais que 1 *drone*, onde cada um identificava a pessoa, confirmava que era uma vítima e seria o mesmo a enviar o alerta para as entidades responsáveis, e entre eles fariam rondas inteligentes de patrulha.

# Bibliografia

[Miguel Griné., 2014] Miguel António Azevedo Griné (2014). *Vessel Detection in Oceanographic Airborne Imagery*.

[Li Liu., 2018] Li Liu, Jie Chen, Paul Fieguth, Guoying Zhao, Rama Chellappa, Matti Pietikainen (2018). *A Survey of Recent Advances in Texture Representation*.

[Igor V. Grigkov 2009] Igor V. Grigkov, Petr P. Koltsov, Nikolay V. Kotovich, Alexander A. Kravchenko, Alexander S. Kutsaev, Andrey S. Osipov, Alexey V. Zakharov (2009). *Performance Evaluation of Texture Segmentation Methods*.

[VoanaBoa, 2018] Regulamento da Utilização de Drones no Espaço Aéreo,  
<http://voanaboa.pt/campanha>

[Python3.2.3, 2012] Python3.2.3 (2012). Python programming language.  
<http://docs.python.org/py3k/>.