

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**UM MÉTODO PARA
IDENTIFICAÇÃO DE
SUPERFÍCIE AQUÁTICA TURVA
PARA NAVEGAÇÃO AUTÔNOMA**

MATEUS EUGÊNIO COLET

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Ciência da Computação na Pontifícia Universidade Católica do Rio Grande do Sul.

Orientadora: Prof. Isabel Harb Manssour

**Porto Alegre
2017**

Ficha Catalográfica

C694m Colet, Mateus Eugênio

Um método para Identificação de superfície aquática turva para navegação autônoma / Mateus Eugênio Colet . – 2016.

95 f.

Dissertação (Mestrado) – Programa de Pós-Graduação em Ciência da Computação, PUCRS.

Orientadora: Profa. Dra. Isabel Harb Manssour.

1. Visão Computacional. 2. Veículo Aquático. 3. Análise de Componentes Principais. 4. Rede Neural Artificial. I. Manssour, Isabel Harb. II. Título.

Elaborada pelo Sistema de Geração Automática de Ficha Catalográfica da PUCRS com os dados fornecidos pelo(a) autor(a).



Pontifícia Universidade Católica do Rio Grande do Sul
FACULDADE DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

TERMO DE APRESENTAÇÃO DE DISSERTAÇÃO DE MESTRADO

Dissertação intitulada "Um Método para Identificação de Superfície Aquática Turva para Navegação Autônoma" apresentada por Mateus Eugênio Colet como parte dos requisitos para obtenção do grau de Mestre em Ciência da Computação, aprovada em 19 de agosto de 2016 pela Comissão Examinadora:

Dra. Isabel Harb Manssour
Orientadora

PPGCC/PUCRS

Dra. Soraia Raupp Musse

PPGCC/PUCRS

Dr. Luiz Marcos Garcia Goncalves

UFRN

Homologada em...../...../....., conforme Ata No. pela Comissão Coordenadora.

Prof. Dr. Luiz Gustavo Leão Fernandes
Coordenador.

AGRADECIMENTOS

Primeiramente agradeço a Deus pelo dom da vida da sabedoria e o poder com que iluminou o meu caminho durante esta caminhada de minha formação.

Agradeço a toda minha família, que me incentivou e apoiou em todos os sentidos em concluir e completar a realização deste trabalho. Em especial à minha mãe, Diva Célia e ao meu pai Arnélio, que além de me oferecer todas as condições para que eu pudesse prosseguir com meus estudos, sempre me incentivaram nas horas difíceis, de desânimo e cansaço. Seus conselhos sempre foram a base e o conforto para superar os obstáculos e buscar a direção da razão e da inteligência para seguir em frente com força e determinação. Aos meus irmãos Héinton e Paulo, que sempre em momentos desvirtuosos fizeram seu papel de irmãos mais velhos, me mostrando com seus ensinamentos e vivência o sentido do correto, da ajuda e da família.

A minha orientadora professora Dr^a. Isabel Harb Manssour, pelos seus ensinamentos, paciência, oportunidade e apoio no auxílio e desenvolvimento deste trabalho, sempre me ensinando de maneira inteligente meus pensamentos, me conduzindo assim a buscar mais conhecimento. A Dr^a. Adriana Braun, pela ajuda e as incansáveis ideias e supervisões ao longo das minhas atividades. Agradeço a elas que durante muito tempo me ensinaram e me conduziram em minha pesquisa, a quais terão meus eternos agradecimentos.

Aos amigos e colegas de mestrado, pelas conversas, ideias, convivências e por me aguentar a maioria do tempo, em que eu sempre os protuberavam com minhas convicções de certezas supérfluas sobre as coisas. Pelas novas amizades que surgiram, nestes anos que com certeza irão permanecer pelo resto da vida.

E claro, a meus antigos amigos de faculdade e de minha cidade natal, que estavam sempre me ajudando, incentivando e torcendo para que tudo desse certo. Amigos que sempre me alegravam com suas festas, churrascos, bagunças e momentos de descontração. E a todos que de alguma forma ou breve momento me ajudaram para a conclusão deste trabalho, agradeço à todos de coração.

UM MÉTODO PARA IDENTIFICAÇÃO DE SUPERFÍCIE AQUÁTICA TURVA PARA NAVEGAÇÃO AUTÔNOMA

RESUMO

A navegação em ambientes aquáticos é um amplo tema de pesquisa que nos últimos anos tem recebido maior atenção da comunidade que trabalha com robótica móvel. O uso de veículos de superfície aquática para inspeção, mitigação e gerenciamento de desastres naturais aquáticos, por exemplo, ajudou a impulsionar a pesquisa para navegação autônoma nesta área. Para executar uma navegação autônoma em ambientes ao ar livre, é necessário identificar partes da superfície que podem ser navegáveis, o que é um dos desafios deste tipo de aplicação. Neste contexto, o objetivo deste trabalho é apresentar um método para identificação da superfície aquática baseada na propriedade óptica turva, encontrada nestes tipos de ambientes. No desenvolvimento deste método foram utilizadas técnicas de visão computacional em conjunto com redes neurais artificiais para construir um classificador, que tem como tarefa distinguir e identificar a superfície aquática navegável. Para isto, foi realizado um estudo sobre o uso de diversas características baseadas na cor e textura das superfícies aquáticas turvas para a extração de diversos atributos para treinar o classificador, tais como: média, entropia, variância e energia, variando em diferentes canais de cor (RGB, HSV, YUV). De forma a compactar todas estas informações foi utilizado o método estatístico de análise de componentes principais, e o seu resultado foi utilizado como entrada das redes neurais artificiais, construindo assim o classificador. Este classificador tem como tarefa fundamental gerar o mapa de navegabilidade que é interpretado por uma máquina de estados para tomada de decisões. Todo o método desenvolvido foi embarcado em um protótipo de veículo aquático ao mesmo tempo em que os resultados e avaliações foram validados utilizando o veículo em ambientes reais e diferente cenários.

Palavras-Chave: Visão Computacional, Veículo Aquático, Análise de Componentes Principais, Rede Neural Artificial.

A METHOD FOR TURBID WATER SURFACE IDENTIFICATION FOR AUTONOMOUS NAVIGATION

ABSTRACT

Navigation in aquatic environments is a broad topic that in recent years has received considerable attention from the community working with mobile robotics. The use of aquatic surface vehicles for inspection, mitigation and natural aquatic disasters management, boosted the search for autonomous navigation in this area. In order to perform an autonomous navigation in outdoor environments, it is necessary to identify parts of the surface that can be navigable, and this is one of the fundamental problems in this type of application. In this context, the objective of this research is to propose a method for water surfaces identification based on the blurred optical property, found in these types of environments. More specifically, computer vision was used in conjunction with neural networks to build a classifier, which has the task of distinguishing and identifying navigable aquatic surface. In order to achieve this objective, a study on the use of several features based on color and texture of these turbid surfaces for the extraction of various attributes to generate the classifier, such as: mean, variance, entropy and energy, varying in different color channels (RGB, HSV, YUV). In order to compress all of this information it was used statistical method of principal component analysis, whose results were used as input of the artificial neural network, thus constructing the classifier. The classifier has the fundamental task of generating the navigation map that is interpreted by a state machine for decision making. All the method developed was applied and embarked in aquatic vehicle prototype at the same time the results and assessments were validated using the vehicle in real environments and different scenarios.

Keywords: Computer Vision, Surface Vehicle, Principal Component Analysis, Artificial Neural Network .

LISTA DE FIGURAS

<p>Figura 1.1 – (a) Números de desastres relacionados ao clima ao redor do mundo [55]; (b) porcentagem de mortos no Brasil por desastre naturais do ano de 1990 a 2012 [5].</p>	15
<p>Figura 2.1 – Cubo de cores RGB (a), espaço de cores HSV cônico (b) e espaço de cores YUV diamante (c).</p>	19
<p>Figura 2.2 – Neurônio biológico (a) e Neurônio artificial (b). As entradas são re- presentadas por $\{x_1, x_2, \dots, x_m\}$ e os pesos são denotados como $\{W_{k1}, W_{k2}, \dots, W_{km}\}$. O símbolo \sum representa o somatório da rede e b_k o bias do neurônio, que tem a finalidade de aumentar ou diminuir o valor da estrada para a função de ativação. V_k é o valor do somador, φ representa a função de transição e Y_k é o valor de saída do neurônio [19].</p>	20
<p>Figura 2.3 – Representação gráfica das topologias das redes neurais artificiais [19].</p>	21
<p>Figura 2.4 – Topologia <i>Multi-Layer Perceptron</i>, com duas camadas intermediárias.</p>	22
<p>Figura 2.5 – Indicativo do ponto ótimo para a parada de treinamento. [34]</p>	23
<p>Figura 3.1 – Relações entre as principais técnicas e métodos apresentadas por Iqbal et al., para detecção de água [23].</p>	25
<p>Figura 3.2 – Resultado da abordagem desenvolvida por Santana et al. [40]. (a) Primeira imagem da sequência de vídeo; (b) O primeiro passo do método refere-se à extração de regiões da imagem de entrada que são susceptí- veis de ser a água, dada a sua assinatura de textura dinâmica (máscara branca); (c) No passo seguinte, uma segmentação baseada na aparência da primeira imagem da sequência de vídeo é processada; (d) Em seguida, esta segmentação é usada para expandir os resultados de detecção em toda a imagem (<i>overlay</i> verde).</p>	27
<p>Figura 3.3 – A água é indiretamente detectada através da detecção do céu pelo reflexão na terra e combinando os resultados [37].</p>	28
<p>Figura 3.4 – Sequência do algoritmo auto supervisionado para extração das ca- racterísticas de classificação e detecção de regiões aquáticas [1].</p>	29
<p>Figura 4.1 – Diagrama com as etapas do processo de identificação da superfície aquática.</p>	33
<p>Figura 4.2 – Imagem de entrada (a) e imagem P subdividida em blocos de tama- nho $B(r, s)$ (b).</p>	34

Figura 4.3 – Imagem de um gráfico <i>scree plot</i> que relaciona o número de componentes D de cada eixo com o valor próprio \hat{r} que lhe está associado [24]	41
Figura 4.4 – Classificador destinado a receber os valores referentes aos CP	42
Figura 4.5 – Exemplo de duas imagens segmentadas manualmente para uso no treinamento. As regiões navegáveis são pintadas de verde e as não navegáveis de vermelho.	45
Figura 4.6 – Modelagem final do classificador. As entradas são componentes da imagem a ser classificada, e a saída é o valor calculado pela classificação.	46
Figura 4.7 – Imagens a serem classificadas (a), obtidas pelo veículo aquático da figura 4.1(a), e mapas de navegabilidade destas imagens classificadas (b), nos quais quanto mais branco maior é a certeza de região navegável.	47
Figura 4.8 – Definição das áreas para as tomadas de decisões (a); Busca das classificações baseada nas áreas pré definidas (b); Combinação entre a áreas pré definidas com o mapa de navegabilidade (c).	49
Figura 4.9 – Diagrama de sequências das ações da MEF.	50
Figura 5.1 – Barco utilizado para a coleta dos dados (a) e um exemplo de imagem coletada do ponto de vista do veículo aquático (b).	52
Figura 5.2 – Gráfico <i>scree plot</i> dos componentes calculados.	55
Figura 5.3 – Características padronizadas a partir de blocos de treinamento projetado no subespaço definido pelos três componentes principais do PCA.	56
Figura 5.4 – Modelagem final da rede neural artificial.	57
Figura 5.5 – Gráfico do valor de erro quadrático médio no treinamento da RNA (a) e valores do método de avaliação S obtidos no treinamento da RNA com 5.000 <i>epochs</i> (b).	58
Figura 5.6 – Comparação de um <i>frame</i> classificado com diferentes treinamentos da RNA: Classificação com treinamento finalizado no <i>epoch</i> 200 (a) e no <i>epoch</i> 5.000 (b).	59
Figura 5.7 – Identificação da linha de horizonte nos <i>frames</i> do vídeo (a) e no respectivo mapa de navegabilidade (b).	63
Figura 5.8 – Definição das classes navegáveis e não navegáveis para o treinamento do segundo classificador	63
Figura 5.9 – Processo do uso de dois classificadores com base na linha de horizonte para a classificação dos <i>frames</i> de entrada.	64
Figura 5.10 – Dois <i>frames</i> que mostram a melhoria no mapa de navegabilidade obtido pela união dos dois classificadores desenvolvidos.	65

Figura 5.11 – Ganho de desempenho nas métricas avaliadas com a utilização dos dois classificadores no cenário I.	66
Figura 5.12 – Conjunto de imagens do cenário III que corresponde a uma situação real de desastre natural. A coluna (a) apresenta as imagens de entrada; (b) Segmentação realizada para métrica de desempenho; (b) O resultado obtido pelo classificador e (d) sobreposição da classificação sobre o conjunto de entrada [38].	69
Figura 5.13 – Média dos resultados das avaliações para os três cenários e seus respectivos erros padrões.	71
Figura 5.14 – Placa Raspberry Pi modelo 2 (a), <i>shield</i> da câmera Pi (b) e adaptação do <i>hardware</i> sobre o veículo aquático (c).	73
Figura 5.15 – Raspberry Pi 2, ambiente executando teste com a biblioteca OpenMP	74
Figura 5.16 – Desempenho alcançado com o uso de OpenMP no hardware Raspberry Pi 2	74
Figura 5.17 – Alguns <i>frames</i> nos quais foi tomada uma decisão correta pelo algoritmo de navegação (a), e incorreta (b), isto é, não de acordo com o esperado.	76
Figura 5.18 – Protótipo desenvolvido em ação em um ambiente real.	77
Figura B.1 – Conjunto de <i>frames</i> utilizados para treinamento.	88
Figura C.1 – Conjunto de <i>frames</i> extraídos para classificação e experimentos. Em (a) apresenta os <i>frames</i> de entrada; (b) Segmentação manual realizada; (c) O resultado obtido; E (d) União da classificação com a imagem de entrada.	90
Figura D.1 – Conjunto de <i>frames</i> reavaliados utilizando o segundo classificador. Aonde (a) apresenta os <i>frames</i> de entrada; (b) Segmentação manual realizada; (c) O resultado obtido; E (d) União da classificação com a imagem de entrada.	92
Figura E.1 – Resultados obtidos na classificação do segundo cenário avaliado em várias tomadas. As imagens em (a) apresentam os <i>frames</i> de entrada; (b) Segmentação manual realizada; (c) O resultado obtido; E (d) União da classificação com a imagem de entrada.	95

LISTA DE TABELAS

Tabela 3.1 – Resumo das informações apresentadas por alguns trabalhos relacionados.....	31
Tabela 5.1 – Tabela de valores do total de variância explicada por componente. .	55
Tabela 5.2 – Tabela de valores e métricas de desempenho do classificador para o cenário I	62
Tabela 5.3 – Tabela de valores e métricas de desempenho com o segundo classificador	65
Tabela 5.4 – Tabela de valores e métricas de desempenho do classificador para o segundo cenário avaliado.	67
Tabela 5.5 – Tabela de valores e métricas de desempenho do classificador, para o teste de variação das imagens de entrada apresentadas pelo cenário III.	70
Tabela 5.6 – Tabela de valores esperados e obtidos na tomada de decisão na navegação.	75
A.1 Tabela contendo os atributos calculados	86

LISTA DE SIGLAS

CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior
CEMADEN – Centro Nacional de Monitoramento e Alertas de Desastres Naturais
CP – Componente Principal
EQM – Erro Quadrático Médio
FANN – Fast Artificial Neural Network Library
FPS – Frames Per Second
HSV – Hue, Saturation, Value
MEF – Máquina de Estados Finitos
MLP – Multi-Layer Perceptron
OPENCV – Open Source Computer Vision Library
OPENMP – Open Multi-Processing
PCA – Principal Component Analysis
RGB – Red, Green, Blue
RNA – Rede Neural Artificial
ROC – Receiver Operating Characteristics
RPI – Raspberry Pi
YUV – luma (Y) and two chrominance (UV)

SUMÁRIO

1	INTRODUÇÃO	14
1.1	MOTIVAÇÃO	15
1.2	OBJETIVO	16
1.3	CONTRIBUIÇÕES	16
1.4	ESTRUTURA DA DISSERTAÇÃO	17
2	FUNDAMENTAÇÃO TEÓRICA	18
2.1	ESPAÇO DE COR	18
2.2	REDES NEURAIS ARTIFICIAIS	19
2.2.1	<i>MULTI-LAYER PERCEPTRON</i>	22
3	TRABALHOS RELACIONADOS	24
3.1	DESAFIOS E PERSPECTIVAS EM AMBIENTES AQUÁTICOS	24
3.2	MÉTODOS DE IDENTIFICAÇÃO USANDO SEGMENTAÇÃO DE IMAGEM ...	25
3.3	MÉTODOS DE APRENDIZADO PARA IDENTIFICAÇÃO DE SUPERFÍCIE AQUÁTICA	27
3.4	COMPARAÇÃO DOS TRABALHOS APRESENTADOS	30
4	DESCRIÇÃO DO MÉTODO	32
4.0.1	METODOLOGIA	32
4.1	DIVISÃO EM BLOCOS	33
4.2	EXTRAÇÃO DAS CARACTERÍSTICAS DAS IMAGENS	34
4.2.1	CÁLCULO DOS ATRIBUTOS DE TEXTURA	35
4.2.2	CONSIDERAÇÕES	36
4.3	SELEÇÃO DAS CARACTERÍSTICAS PARA GERAR O CLASSIFICADOR	37
4.3.1	<i>PRINCIPAL COMPONENT ANALYSIS</i>	37
4.3.2	CONSIDERAÇÕES	42
4.4	CLASSIFICADOR	42
4.4.1	AVALIAÇÃO DO CLASSIFICADOR	43
4.4.2	PROCESSO DE TREINAMENTO DO CLASSIFICADOR	44
4.4.3	MODELAGEM FINAL DO CLASSIFICADOR	45
4.4.4	CLASSIFICANDO NOVAS IMAGENS	47
4.5	NAVEGAÇÃO	48

4.5.1	CONSIDERAÇÃO FINAL	50
5	TESTE E RESULTADOS	52
5.1	DEFINIÇÃO DOS PARÂMETROS INICIAIS	53
5.1.1	REDUÇÃO DA DIMENSIONALIDADE POR ANÁLISE DE COMPONENTE PRINCIPAL	54
5.2	REDE NEURAL E AVALIAÇÕES	56
5.2.1	TREINAMENTO	57
5.2.2	CONSIDERAÇÕES	59
5.3	CLASSIFICAÇÃO	59
5.3.1	CENÁRIO I.....	61
5.3.2	LINHA DO HORIZONTE E SEGUNDA REDE NEURAL	62
5.3.3	CENÁRIO II	65
5.3.4	DIVERSIFICANDO A ENTRADA	66
5.3.5	ANÁLISE DOS RESULTADOS E CONSIDERAÇÕES.....	70
5.4	EMBARCANDO O MÉTODO	71
5.4.1	HARDWARE	72
5.4.2	TOMADA DE DECISÃO NA NAVEGAÇÃO	75
5.4.3	PROTÓTIPO	76
6	CONCLUSÃO	78
6.1	TRABALHOS FUTUROS	79
	REFERÊNCIAS	81
	APÊNDICE A – Total de características extraídas para geração dos atributos .	86
	APÊNDICE B – Coleção de frames de treinamento	88
	APÊNDICE C – Resultados obtidos com a primeira classificação	89
	APÊNDICE D – Resultados obtidos com a nova classificação do cenário I, utilizando o segundo classificador	91
	APÊNDICE E – Resultados obtidos na classificação do segundo cenário	93

1. INTRODUÇÃO

Com os recentes avanços tecnológicos, diversas áreas do conhecimento têm se beneficiado com técnicas de processamento digital de imagens e visão computacional. Visão computacional segundo Forsyth [15] é a extração e descrição do mundo a partir de imagens ou sequencias de imagens. Mallot [28], caracteriza a visão computacional em seu livro como "ótica inversa", ou seja, como o processamento de informações visuais, resulta em uma descrição do ambiente. Mas de forma geral a visão computacional consiste na ciência e na tecnologia de como as máquinas enxergam o ambiente ao seu redor [41], e suas aplicações se estendem a várias áreas da computação, como robótica, sistemas de automação, veículos autônomos entre outras.

A área da robótica, principalmente, destaca-se pela vasta utilização de algoritmos da área de visão computacional com o objetivo de dar conhecimento ao agente do universo ao seu redor, pois, além de usar sensores como forma de obtenção de informação, a visão é de grande valor pela quantidade de dados que podem ser fornecidos [23]. Sendo assim, neste contexto, a visão computacional pode ser aplicada nos mais variados tipos de ambientes, tais como: terrestres, através de carros autônomos e robôs humanoides; aéreos, através de drones e quadricópteros; e aquáticos, com veículos de superfície e submersos [39].

Ambientes aquáticos fornecem casos interessantes de aplicações robóticas, entre os quais se destacam: funções de monitoramento para realizar a segurança e missões batimétricas [9]; exploração de oceano [29]; e monitoramento de ambientes marinhos [56] ou ambientais [11]. Além disso, destacam-se também por serem aplicados a desastres naturais como mitigação de inundações [43, 53, 43], ou sendo cooperativo com outros tipos de veículos [32].

Ao explorar esse tipo de ambiente o reconhecimento da superfície aquática se torna muito importante, pois é um método de entrada de informação para o veículo aquático ter o conhecimento necessário para um deslocamento seguro e preciso [23]. Normalmente a capacidade de reconhecimento da superfície é utilizada como pré-requisito para que outras tarefas mais complexas possam ser executas.

A navegação em ambientes naturais é, certamente, um problema complexo [23]. As maiores dificuldades são as características dinâmicas que as superfícies aquáticas apresentam devido às várias dependências das imagens como brilho e saturação de cor. Fatores físicos como intensidade luminosa, sombras, reflexos, e efeitos de difração e refração, influenciam diretamente no processo de identificação para a navegação [41].

1.1 Motivação

As organizações internacionais relacionadas à redução de riscos, apresentam estatísticas que os impactos causados por inundação afetam mais de 500 milhões de pessoas, gerando um custo de 50 milhões de dólares ao ano e com o número de mortes mais alto registrado em desastres naturais, ultrapassando 1.5 milhões desde o início do século [25]. A Figura 1.1(a) apresenta um infográfico que mostra um aumento no número de desastres mundiais relacionados a fenômenos pluviométricos nas últimas décadas. A Figura 1.1(b), apresenta estudos nacionais recentes, que mostram que no ano de 2012 no Brasil a enxurrada é o tipo de desastre que causou o maior número de mortes, 58,15% do total [5, 10]. Dos desastres catalogados no período de 1990 a 2012, 22% ocorreram nos anos de 2010 a 2012, ligados a mudanças climáticas. Infelizmente, os desastres de inundação causam maiores efeitos às pessoas nos países subdesenvolvidos, devido à falta de sistemas de alerta precoce, controle de inundação e de uma infraestrutura que possibilite uma rápida resposta em situações de emergência [43].

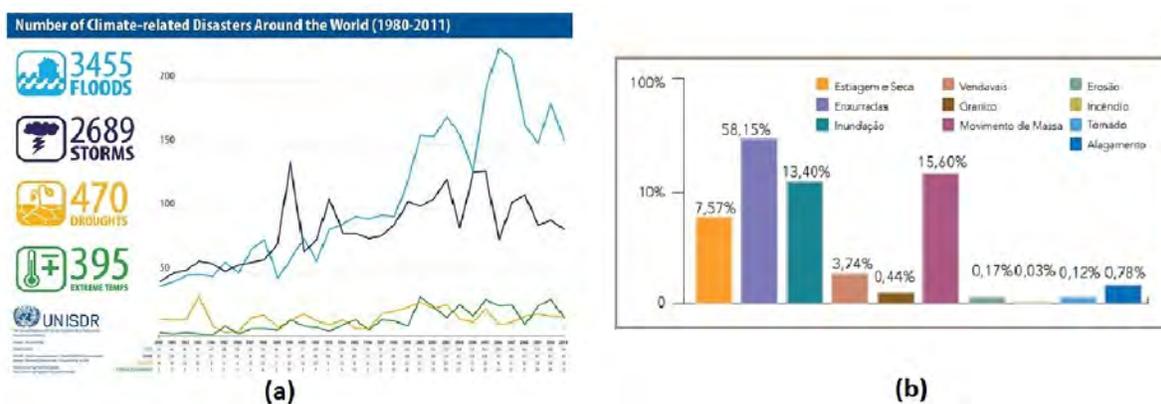


Figura 1.1 – (a) Números de desastres relacionados ao clima ao redor do mundo [55]; (b) porcentagem de mortos no Brasil por desastre naturais do ano de 1990 a 2012 [5].

Considerando este contexto, surgiu a seguinte questão de pesquisa: que tipo de informações podem ser coletadas das superfícies aquáticas turvas usando visão computacional, que aplicadas a um veículo autônomo podem auxiliá-lo na navegação neste ambiente? Neste trabalho, superfície aquática turva refere-se à turbidez da água, ou seja, à presença de partículas sólidas em suspensão na superfície aquática, alterando as suas propriedades ópticas. Portanto, pensando em um veículo aquático inserido em um ambiente pós-desastre, foi desenvolvido um método para reconhecimento da superfície aquática turva considerando as propriedades ótica e física presentes na água e usando a técnica de aprendizado de máquina conhecida como redes neurais. Este método foi usado para detectar áreas ou regiões possíveis de navegabilidade em superfícies aquáticas e aplicar na navegação autônoma.

1.2 Objetivo

O objetivo deste trabalho foi de propor, desenvolver e avaliar um método de identificação de superfícies aquáticas turvas navegáveis, buscando concentrar no subproblema chave da segmentação automática para navegação autônoma. Em outras palavras, a finalidade foi projetar e desenvolver um método capaz de identificar e fazer um mapeamento de uma região navegável baseando-se principalmente nas informações visuais obtidas a partir de uma câmera monocular frontal posicionada sobre um veículo aquático. Mais especificamente, foram utilizados métodos estatísticos em conjunto a um método de inteligência artificial com aprendizado de máquina supervisionado conhecido como redes neurais artificiais, buscando resolver o problema de pesquisa. Foi proposto também, incorporar o método desenvolvido a um sistema embarcado para navegação, podendo ser um barco, uma plataforma navegável ou um dispositivo de menor porte.

O desenvolvimento do método de navegação proposto está inserido em um projeto mais amplo, que visa o desenvolvimento e integração de veículos autônomos inteligentes aplicados na prevenção e mitigação de desastres naturais. Este projeto está em desenvolvimento em parceria com várias universidades do Brasil, e foi elaborado para o edital Pró-Alertas n° 24/2014 da CAPES, que visa estimular no país a realização de projetos conjuntos de pesquisa, com vistas a contribuir para a consolidação do Centro Nacional de Monitoramento e Alertas de Desastres Naturais (CEMADEN).

1.3 Contribuições

As principais contribuições com a realização do trabalho proposto são:

- a proposta de um método para estimar limites de superfícies aquáticas turvas navegáveis a partir de imagens capturadas por uma câmera monocular frontal localizada sobre um veículo aquático em relação ao plano de região navegável;
- geração de um mapa de navegabilidade que determina os limites de regiões navegáveis e que pode ser interpretado por outros métodos ou algoritmos para navegação;
- apresentação de um algoritmo para tomada de ações com base em um mapa de navegabilidade, com o objetivo de tornar autônoma uma navegação em ambientes aquáticos.

1.4 Estrutura da Dissertação

Esta dissertação está organizada da seguinte maneira: no capítulo 2 é apresentada uma fundamentação teórica sobre alguns temas utilizados na descrição do método apresentado. O capítulo 3 contém um resumo e uma análise de alguns trabalhos relacionados, que utilizam métodos e técnicas para segmentação ou identificação da superfície aquática. O capítulo 4, descreve a metodologia proposta para a identificação da superfície aquática, incluindo as suas quatro etapas principais: método de divisão em blocos, extração das características das imagens, método estatístico para redução de dimensionalidade e a construção do método de aprendizado supervisionado usando rede neural artificial. O capítulo 5, apresenta os experimentos e resultados alcançados, com uma avaliação de cada etapa e resultado obtido. Por fim, no capítulo 6, é descrita a conclusão deste trabalho e as próximas etapas para aperfeiçoamento e continuação do método proposto.

2. FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo, é de apresentar a fundamentação teórica necessária para caracterizar os conceitos principais da descrição do método apresentado. Deste modo, são apresentados alguns conceitos necessários para o entendimento e detalhamento de cada etapa da construção do método proposto.

2.1 Espaço de cor

Um espaço de cor possui uma especificação em um sistema de coordenadas tridimensionais. Neste sistema de coordenadas, cada cor é representada por um único ponto desse espaço tridimensional [50, 17].

No espaço RGB, cada cor aparece nos seus componentes espectrais primários de vermelho (*Red*), verde (*Green*) e azul (*Blue*). Baseado em um sistema de coordenadas cartesianas, tem como subespaço de cores de interesse um cubo. Na figura 2.1(a), está representado o cubo RGB, cuja a origem consiste na cor preta e cada eixo corresponde às cores primárias ou seja, vermelho, verde e azul. A combinação de vermelho e verde gera o amarelo, azul e verde gera o ciano e vermelho e azul forma o magenta. A intensidade máxima dos três componente primário, corresponde a cor branca, como pode ser observado nos cantos do cubo RGB apresentado na figura 2.1(a) [42].

A matiz (*Hue*) é um atributo que descreve o comprimento de onda dominante de uma cor, enquanto a saturação (*Saturation*) fornece uma medida do grau de diluição de uma cor pura por luz branca. O espaço HSV de cores possui duas características principais: O componente de intensidade (*Value*) é desacoplado da informação de cor da imagem e os componentes de matiz e saturação são relacionados à percepção humana de cores. A representação gráfica tridimensional do modelo HSV é uma pirâmide hexagonal derivada do cubo RGB, com um ângulo entre os vértices de 60°. O vermelho corresponde a 0°, o amarelo a 60°, o verde a 120°, assim por diante, como apresentado na figura 2.1(b). As diferentes matizes estão representadas na parte superior da pirâmide, a saturação é medida ao longo do eixo horizontal e a luminância é medida ao longo do eixo vertical, o qual passa pelo centro da pirâmide.

O espaço cromático YUV é um sistema de cor baseado no par, luminância/ crominância. Ele separa a luminosidade de uma dada informação cromática para determinar sua matiz. A figura 2.1(c) apresenta o espaço de cor YUV em forma de diamante. Ele armazena a luminosidade de um pixel e um canal de luminância (Y) e dois componente de crominância: (U) a diferença entre a intensidade no canal de azul e a luminância, e (V) a diferença entre a intensidade no canal de vermelho e a luminância. A importância da utilização do sis-

tema de cores YUV é que a luminância e a informação de cor são independentes. Imagens tendo apenas componente do espaço (Y) sem qualquer (U) e (V), são apenas imagens em escala de cinza que variam do preto para o branco [57]. Nenhum dos modelos referidos anteriormente tem em conta uma propriedade de visão humana. Esta é mais sensível as mudanças de intensidade da luz (luminância) do que da cor (crominância).

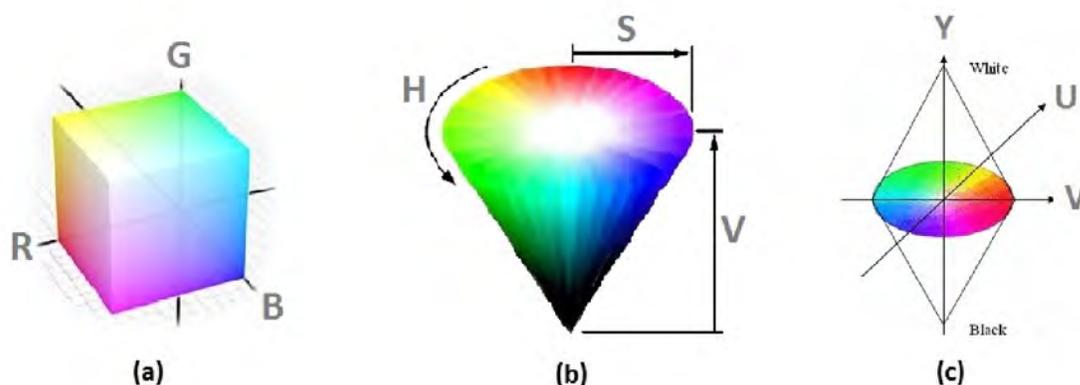


Figura 2.1 – Cubo de cores RGB (a), espaço de cores HSV cônico (b) e espaço de cores YUV diamante (c).

2.2 Redes Neurais Artificiais

Redes neurais artificiais são sistemas computacionais constituídos por unidades conhecidas como neurônios. Os neurônios são elementos interconectados responsáveis pelo processamento das informações, que trabalham em paralelo para desempenhar uma determinada tarefa [2]. A Figura 2.2(b) apresenta a estrutura de um neurônio artificial inspirado na funcionalidade dos neurônios biológicos (Figura 2.2(a)). Portanto, as RNAs são dispositivos não-lineares que buscam resolver os problemas de forma análoga ao cérebro humano, simulando computacionalmente um conjunto de neurônios e conexões que fazem o processamento da informação. As RNAs geralmente são construídas e aplicadas no reconhecimento de padrões, na otimização e na previsão de sistemas computacionais complexos [35, 19].

Os modelos de RNAs constituem uma importante técnica estatística não linear capaz de resolver uma gama de problemas de grande complexidade, tornando-se um método extremamente útil em situações em que não é possível definir explicitamente uma sequência de regras [35]. Em geral, isso acontece quando o ambiente dos dados de entrada tem uma imprevisibilidade ou muda constantemente, tornando as RNAs uma boa solução para aplicação em áreas de classificação de padrões e previsão [4, 19]. A pesquisa em redes neurais é bastante ampla e inclui áreas da robótica, sistemas de produção, reconhecimento

de padrões, processamento de imagens, engenharia, biomedicina, análise financeira, gerenciamento de pesquisa operacional entre outras.

Duas vantagens proporcionadas pelas RNAs e que fazem sua aplicação extremamente disseminada, são a flexibilidade e o desempenho [49]. A habilidade de manipular dados imprecisos faz com que as redes neurais sejam extremamente eficazes em tarefas nas quais sistemas especialistas não estão à disposição ou quando um conjunto de regras ou sua alteração deve ser formulada de acordo com o esperado. Uma rede neural consegue lidar com pequenas mudanças na informação de entrada, tais como sinais ruidosos ou outras alterações. Por exemplo, se a informação de entrada é a imagem de um objeto, a correspondente resposta mantém-se inalterada, mesmo se a imagem muda um pouco de brilho ou o objeto muda um pouco de formato. O desempenho é bastante atrativo e vantajoso, devido a sua estrutura ser paralela. A implementação e execução de uma RNA em computadores, hardwares ou dispositivos eletrônicos especiais permite obter respostas em tempo real [2].

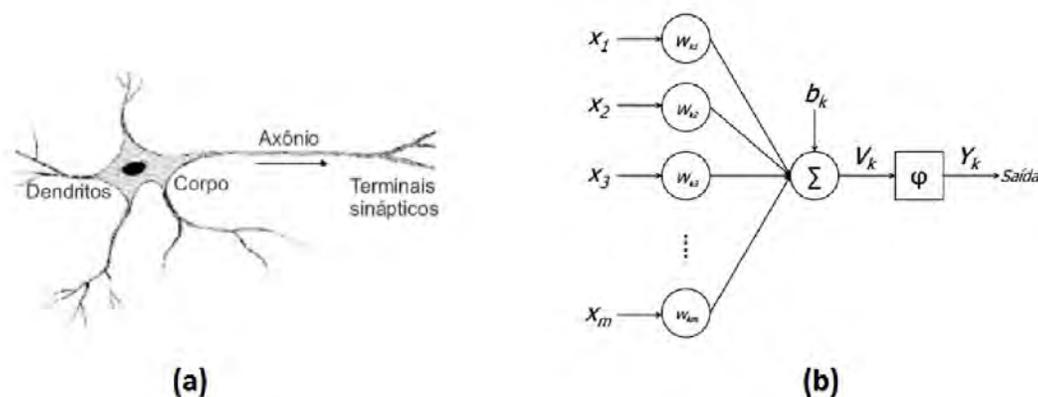


Figura 2.2 – Neurônio biológico (a) e Neurônio artificial (b). As entradas são representadas por $\{x_1, x_2, \dots, x_m\}$ e os pesos são denotados como $\{W_{k1}, W_{k2}, \dots, W_{km}\}$. O símbolo Σ representa o somatório da rede e b_k o bias do neurônio, que tem a finalidade de aumentar ou diminuir o valor da estrada para a função de ativação. V_k é o valor do somador, φ representa a função de transição e Y_k é o valor de saída do neurônio [19].

A forma de aprendizado de uma RNA é uma característica muito importante. Basicamente existem duas formas de aprendizado: o supervisionado e o não supervisionado. No método supervisionado é fornecido um conjunto de entradas e saídas esperadas para o treinamento e a rede ajusta os seus pesos (sinápticos) com base nestas informações. Já no método não supervisionado não são informadas as saídas esperadas, que fornecem os resultados previstos para um conjunto de entradas, assim a rede deve buscar associações relevantes a partir da extração das propriedades estatísticas apenas com os dados de entrada, criando classes e grupos representativos [34].

Através do processo de treinamento, as redes neurais passam a ser capazes de responder a estímulos de entrada. De acordo com cada treinamento de aprendizado, a rede

torna-se capaz de reconhecer padrões, classificando as entradas de forma que, quando uma entrada ou um conjunto de entradas é aplicado à rede, esta fornece uma resposta de saída indicando a qual classe a entrada pertence. Numa outra forma de aprendizado, a rede aprende sobre o relacionamento que há entre pares entrada-saída, assim quando uma entrada for aplicada à rede, esta responde com a saída correspondente àquela entrada [19].

As configurações de como esses neurônios estão conectados influenciam diretamente na forma como as RNAs são treinadas e, conseqüentemente, no funcionamento da rede e como são esperados os dados de saída [19]. Assim, várias formas de organizar os neurônios foram sistematizadas, criando o conceito de topologia das redes neurais artificiais. A figura 2.3 apresenta os três tipos de topologia de redes neurais: *Perceptron*, *Multi-Layer Perceptron* e *Recorrente*.

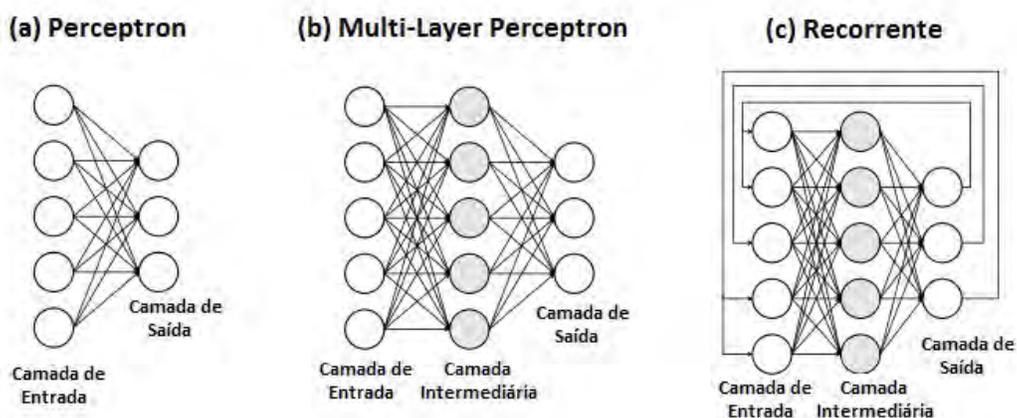


Figura 2.3 – Representação gráfica das topologias das redes neurais artificiais [19].

A definição da topologia determina algumas características presentes na rede, tais como a quantidade de neurônios, a necessidade e quantidade de camadas intermediárias (*Multi-layer Perceptron*) ou a forma cíclica ou retro-alimentada (*Recorrente*). A topologia ou arquitetura das RNAs, são definidas para favorecer aplicações com certo domínio específico, gerando uma alternativa para soluções que possuem algumas características individuais. Ou seja, a escolha da RNA mais apropriada depende do problema a ser resolvido.

Resumidamente, a topologia *Perceptron*, figura 2.3(a), conhecida também como *single-layer feedforward*, é a configuração mais simples de RNA. Possui apenas camadas de entradas e de saídas, sem ciclos entre as conexões e os neurônios, ficando limitada à solução de problemas lineares. Na topologia *Recorrente*, figura 2.3(c), as saídas são conectadas às entradas da rede, realimentando a RNA, e podem ter indeterminadas camadas intermediárias. Esta configuração de rede recorrente possui a capacidade de simular uma memória de curto prazo [19, 2]. A topologia *Multi-layer Perceptron* (MLP), figura 2.3(b), é similar a *Perceptron*, mas com a possibilidade de adicionar camadas intermediárias, isso

torna o modelo capaz de resolver problemas não lineares e apresenta melhor precisão nos resultados da RNA. Como esta foi a topologia utilizada para a construção da RNA deste trabalho, um detalhamento das redes MLP é feito na subseção a seguir.

2.2.1 *Multi-Layer Perceptron*

A construção de uma rede MLP apresenta no mínimo três camadas de neurônios, sendo uma camada de entrada, uma de saída e no mínimo uma camada intermediária (ou oculta) [19]. A figura 2.4, apresenta a configuração de uma rede MLP, em que os dados de entrada são propagados através das conexões, passando pelas camadas intermediárias da rede até a camada de saída. As conexões de cada neurônio são conectadas a todos os neurônios da camada seguinte.

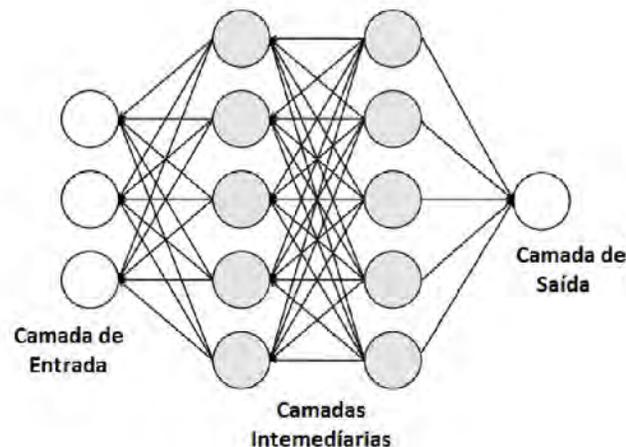


Figura 2.4 – Topologia *Multi-Layer Perceptron*, com duas camadas intermediárias.

Basicamente, o processo de execução de uma rede MLP consiste em realizar inicialmente a coleta dos dados. Os dados que serão inseridos na rede devem ser dados numéricos e, dependendo da aplicação, esses dados devem ser codificados. Por fim, os dados devem ser divididos em conjuntos ou amostras, em que parte desses dados são usados para o treinamento da rede e validação, e outra parte para verificação. Redes MLP são redes artificiais *feedforward*, modelo que propaga progressivamente o sinal dos conjuntos de entrada para a saída apropriada. O vetor de dados de entrada é inserido na camada de entrada da rede, e os dados são processados pelos neurônios dessa camada e são propagados para a camada seguinte. O processo se repete a cada camada até atingir a camada de saída [19]. Os sinais propagados são chamados de sinais funcionais. Para esse processo *feedforward* os pesos permanecem estáticos. Redes MLP também podem ser treinadas usando *backpropagation*, em que é possível mudar os pesos sinápticos da

rede depois de cada conjunto de dados ser processado com uma regra de correção de erros. Um dos critérios é encerrar o treinamento após o erro médio quadrático ficar abaixo de um valor α pré-definido. Após o cálculo do critério definido, o valor a ser ajustado percorre o caminho inverso da rede ajustando o peso dos neurônios [49]. *Backpropagation* é uma etapa em que se propaga o sinal, chamado de sinal de erro.

No treinamento *backpropagation* de uma rede MLP, dois parâmetros são importantes para a qualidade do resultado: a taxa de aprendizado e o *momentum* [49]. O parâmetro taxa de aprendizado tem grande influência durante o processo de treinamento da rede neural. Uma taxa de aprendizado muito baixa torna o aprendizado da rede muito lento, ao passo que uma taxa de aprendizado muito alta provoca oscilações no treinamento e impede a convergência do processo de aprendizado. O parâmetro *momentum* é um valor entre [0.0, 1.0], que tem por objetivo aumentar a velocidade de treinamento da rede e reduzir o perigo de instabilidade. A utilização destes parâmetros é importante para determinar o encerramento do treinamento.

O critério de parada de treinamento é feito principalmente pela validação cruzada, quando o resultado ótimo é encontrado. A figura 2.5 apresenta graficamente quando o treinamento deverá ser interrompido visando uma melhor capacidade de generalização da rede neural, indicando que o treinamento produziu melhoria no resultado até o ponto do resultado ótimo [34]. Mas, também pode-se definir o critério de parada utilizando um número de *epochs* ou ciclos de treinamento limite. Ou seja, no caso da validação cruzada, se o treinamento não parar, o mesmo será finalizado ao atingir o número de *epochs* estabelecido. Deste modo, a rede finaliza seu treinamento sempre no critério de limite de *epochs*, o que é um indicativo que os ajustes dos pesos sinápticos na rede não estão produzindo bons resultados, indicando um sub-ajustamento ou super-ajustamento.

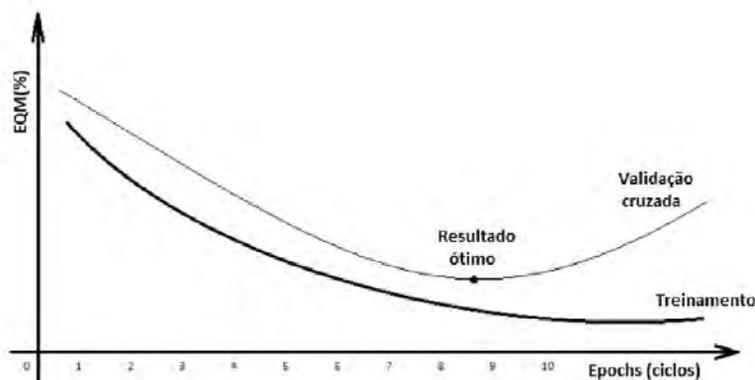


Figura 2.5 – Indicativo do ponto ótimo para a parada de treinamento. [34]

3. TRABALHOS RELACIONADOS

Neste capítulo são apresentados alguns trabalhos que abordam a identificação de superfície aquática, através dos quais é possível perceber os desafios e as perspectivas de se trabalhar em tais ambientes. Além disso, são descritos alguns métodos, modelos e abordagens que, a partir de imagens de uma câmera, permitem a identificação ou locomoção nestes ambientes usando técnicas de visão computacional, processamento de imagens e aprendizado de máquina.

Sendo assim, este capítulo está organizado da seguinte maneira: A seção 3.1 apresenta alguns trabalhos que abordam os desafios e perspectivas de se trabalhar em ambientes aquáticos, dos fundamentos até as metodologias utilizadas para a detecção de superfície aquática em diferentes condições e ambientes; A seção 3.2, aborda a identificação baseada em visão com trabalhos que usam extração de características das imagens para obter dados de intensidade, saturação, brilho e espaço de cor de grandes conjuntos de pixels; A seção 3.3, descreve trabalhos que usam aprendizado de máquina supervisionado em ambientes aquáticos, aplicado em sistemas de detecção ou navegação de veículos autônomos para estes ambientes; finalmente, a seção 3.4, apresenta as considerações e comparações sobre os trabalhos relacionados em comparação ao método apresentado no capítulo 4.

3.1 Desafios e perspectivas em ambientes aquáticos

Diversos trabalhos têm sido conduzidos na área de visão computacional aplicada a métodos de identificação de superfície aquática, por seu interesse tanto comercial como científico. Deste modo, alguns autores [60, 23] abordam a identificação da superfície aquática, mas cada um segue uma linha de pesquisa diferente.

Uma descrição detalhada das consequências, desafios e influências e variações das variáveis que afetam a capacidade de detectar superfícies aquáticas por meios ópticos, é feita em um amplo trabalho realizado por Iqbal et al. [23]. Este trabalho foca nas dificuldades envolvidas na detecção de corpos de água, juntamente com o estado da arte e técnica que tratam deste tema. Andrew et al. [6], também enfatizam que ambientes aquáticos apresentam vários problemas, tal como a questão da reflexão de outros objetos na superfície da água. Existem também as correntes aquáticas e ondas que distorcem o aspecto da água, ou até mesmo a presença de detritos ou sedimentos que mudam a a sua cor ou deslocam-se na sua superfície.

Considerando estes problemas, cada pesquisador usa uma técnica ou abordagem diferente para contorná-los e obter o resultado desejado dentro das grandes opções de dire-

ção a serem seguidas em ambientes naturais aquáticos, que, em geral, envolve a detecção de regiões de água. Neste sentido, o trabalho de Iqbal et al. [23] faz um levantamento sobre a identificação de superfícies aquáticas, que direciona uma série de métodos para selecionar a técnica e o conjunto de ferramentas mais indicado de acordo com cada situação e imagem. A figura 3.1 apresenta esta tabela, relativa aos trabalhos relacionados de Iqbal et al.

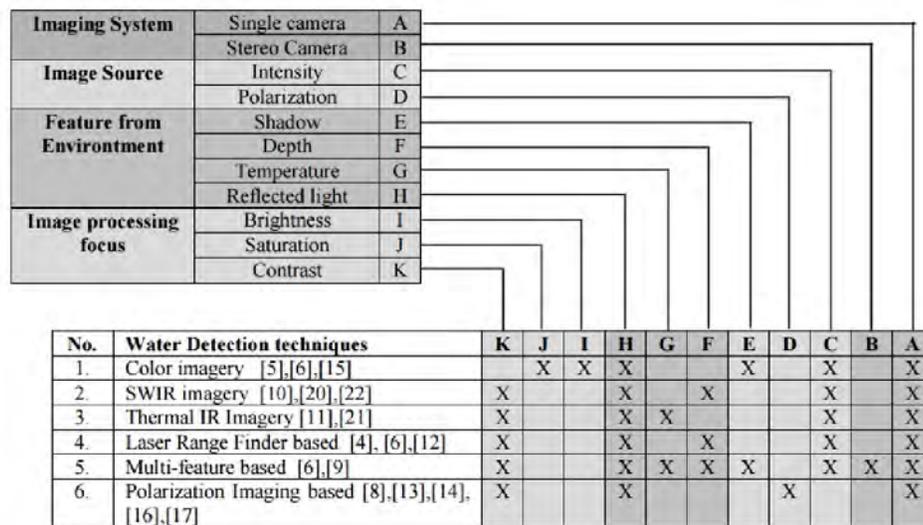


Figura 3.1 – Relações entre as principais técnicas e métodos apresentadas por Iqbal et al., para detecção de água [23].

3.2 Métodos de identificação usando segmentação de imagem

Trabalhos que abordam o uso de visão computacional e segmentação de imagem são os mais disseminados na área de identificação de superfície aquática. De acordo com, Huntsberger et al. [22] e Yao et al. [60], um veículo equipado com um detector de água baseado em visão computacional tem uma maior probabilidade de navegar com segurança e eficientemente. Isto é particularmente enfatizado quando o veículo está em um ambiente desconhecido. O uso de visão computacional é adotado tanto na identificação como na navegação, e com métodos de processamento de imagem torna-se bastante atrativo e disseminado em navegação autônoma aquática [51].

Por exemplo, o trabalho apresentado por Snyder et al. [51] ilustra a navegação de um veículo autônomo que mapeia em tempo real a superfície da água usando técnicas de visão computacional. As imagens de entrada são obtidas a partir de um conjunto circular de câmeras. Através de um algoritmo que detecta os movimentos não rígidos coletados

nas imagens de entrada, os movimentos não rígidos são os movimentos de curta duração das pequenas ondas sobre a superfície aquática, estas movimentações são consideradas como sinais de presença de água. Resumidamente, a segmentação da imagem começa pela determinação da textura de cor que fornece uma probabilidade maior de identificar uma região de água. Na sequência, são inseridos pontos (*cornes*) aleatórios na região sobre a possível área, para posterior aplicação de um rastreador de fluxo de movimento nesta região.

Seguindo o mesmo raciocínio o trabalho de Santana et al. [40] apresenta um modelo de detecção de água para sequências de vídeo, com segmentação de texturas dinâmicas. É definido que a região de água possui uma assinatura, e assim é calculada a entropia a partir do fluxo óptico obtido através de vários *frames*. Para promover uma classificação das regiões com um maior grau de confiabilidade, é aplicado mais um método de propagação de rótulo de segmentação guiada baseado em aparência. Em suma, este processo rótula toda imagem em segmentos como água se estes abrangem uma determinada percentagem de pixels já classificados como água pelo método baseado na entropia e textura. Por fim, é aplicada uma máscara da segmentação como resultado final. A figura 3.2 apresenta o resultado final de todo o processo.

A visão monocular é adequada quando envolve pequenos veículos como barcos, lanchas ou drones, devido ao fato de serem de fácil adaptação, pequenas e os veículos terem limitações de tamanho e capacidade de carga e operar em ambientes adversos como ambientes aquáticos [40]. Alguns autores [6, 59] adotam esta alternativa para a detecção de água, por exemplo, desenvolvendo trabalhos que apresentam visão monocular para uma aeronave que possui a habilidade de voar alguns metros acima da água de forma autônoma para explorar e mapear um rio, a vegetação ou dados específicos. Para auxiliar no problema da identificação e navegação, os veículos também podem utilizar outros sensores, como laser e GPS, que formam um conjunto adicional de dados de entrada que junto com a visão computacional fornecem uma confiabilidade maior na criação de um modelo ou técnica [6].

A definição do uso de várias características para identificação da superfície aquática se baseou em diversos trabalhos [36, 1, 23], os quais indicam que esta superfície sofre alteração de sua propriedade ótica, como saturação e brilho, que ao mesmo tempo não é uniforme, fazendo com que a superfície da água tenha alterações de variação de cor. Algumas técnicas se mostraram robustas ao segmentar um corpo de água com análise de cor, usando vários espaços de cor, tais como RGB e HSB e Lab, afirmando que um descritor de cor associado com um vetor de características se torna robusto utilizando os espaços de cores individualmente e em várias combinações para formar classificadores [36].

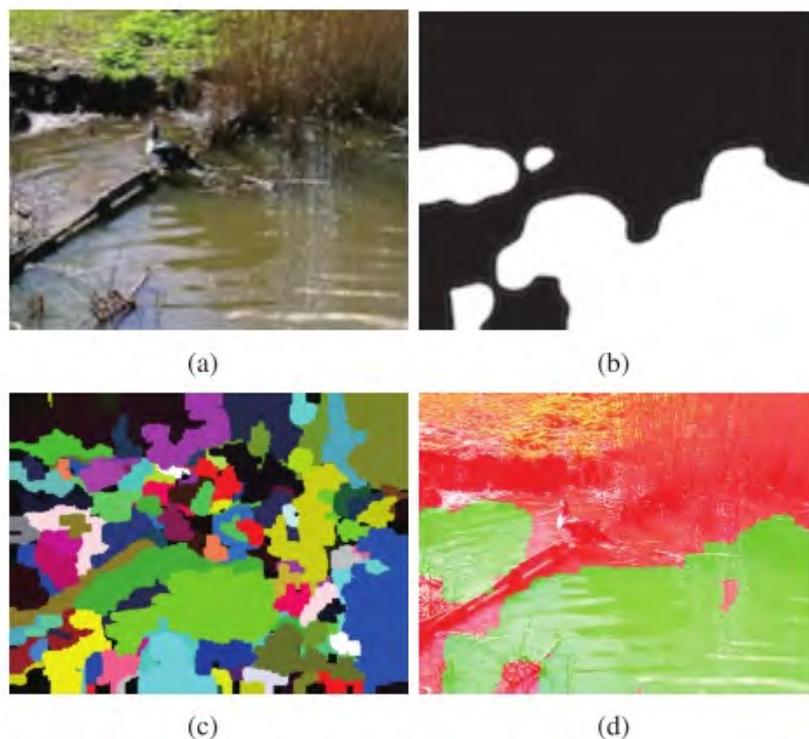


Figura 3.2 – Resultado da abordagem desenvolvida por Santana et al. [40]. (a) Primeira imagem da sequência de vídeo; (b) O primeiro passo do método refere-se à extração de regiões da imagem de entrada que são susceptíveis de ser a água, dada a sua assinatura de textura dinâmica (máscara branca); (c) No passo seguinte, uma segmentação baseada na aparência da primeira imagem da sequência de vídeo é processada; (d) Em seguida, esta segmentação é usada para expandir os resultados de detecção em toda a imagem (*overlay verde*).

3.3 Métodos de aprendizado para identificação de superfície aquática

Um método um pouco mais complexo para veículos aquáticos, capaz de reconhecer mais características, é através do uso de classificadores de aprendizagem de máquina. Assim, a partir de um conjunto de dados e um treinamento, é possível distinguir superfícies aquáticas.

No trabalho desenvolvido por Gong et al. [16], é apresentado um algoritmo de duas fases para encontrar a margem entre a água e a terra, empregando múltiplas características. As imagens coletadas são classificadas em dois tipos: reflexão não identificável e reflexão identificável. Com base nessa classificação, as imagens são posteriormente analisadas com técnicas adequadas, para ambos os casos. Primeiro a imagem é segmentada em regiões menores com base na cor e homogeneidade. Depois as regiões segmentadas são classificadas em áreas de terra ou de água de acordo com características como simetria e brilho. O algoritmo consiste em traçar uma borda, com o objetivo de separar as regiões

de água das regiões de terra por meio de um classificador usando um limiar adaptativo com segmentação das duas categorias.

No entanto, classificadores com base na aparência da água consistem em uma abordagem tendenciosa, devido a reflexões impostas pela vegetação, céu, estruturas, sombras e ondulações na superfície da água. Estes são alguns dos problemas encontrados no meio aquático, que tornam os resultados não tão satisfatórios. Porém, há trabalhos que usam essa informação como uma forma de coleta de informação para distinção. O trabalho de Rankin et al. [37], por exemplo, detecta corpos de água considerando a variação de cor. A proposta deste trabalho é usar as reflexões do céu pelo pressuposto que a superfície da água atua como um espelho horizontal de grande incidência. Como mostra a figura 3.3, o detector baseado em reflexões do céu, geometricamente localiza o pixel no céu que conseqüentemente irá se refletir sobre um pixel na água que terá uma probabilidade de ser o mesmo pixel, para assim identificar a superfície aquática.

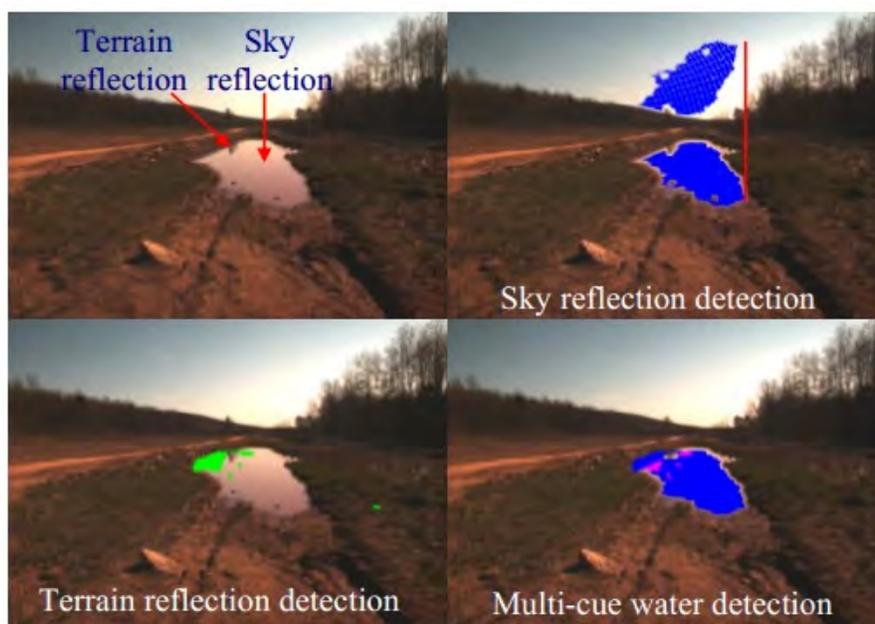


Figura 3.3 – A água é indiretamente detectada através da detecção do céu pelo reflexão na terra e combinando os resultados [37].

Um aprendizado auto supervisionado de classificadores em detecção de água também é proposto por Achar et al. [1] e Heidarsson e Sukhatme [20]. A ideia geral destes trabalhos consiste em usar conhecimentos geométricos com um sistema que é treinado com base em uma *priori* das estruturas do ambiente real [1]. Como alternativa pode ser usado um outro sensor, como um sonar subaquático [20], para a marcação automática do que é água ou não, através de amostras de informação visual, possibilitando, assim, a geração de conjuntos de treinamento on-line de recursos, em que, ao mesmo tempo que é realizada a classificação, algumas amostras são utilizadas para realimentar o aprendizado.

A figura 3.4, apresenta as seguintes etapas de classificação do algoritmo auto-supervisionado desenvolvido por Achar et al. [1]: (a) imagem de entrada para o algoritmo; (b) e (c) árvores de dependência, construídas uma acima e outra abaixo da linha do horizonte da imagem; (d) Cores quentes são usadas para valores próximos a 1 (região costeira), e a região do céu é marcada com preto; (e) exemplos de treinamento: O algoritmo encontra regiões da costa (sombreadas verde) e regiões candidato do rio (sombreado vermelho) para a formação de um classificador; (f) Resultado final. Nas árvores de dependência os nós R, G e B correspondem ao espaço de cor RGB. L, a e b são os nós que correspondem ao espaço de cor Lab. T é o nó característico das texturas (norma do descritor de textura). As arestas ligam características bem correlacionadas, por exemplo acima da linha do horizonte por causa da vegetação, a árvore tem ligações com o espaço de cor G. Informações como simetria podem ser usadas para melhorar o processo de classificação através da redução dos efeitos de reflexões na água [20].

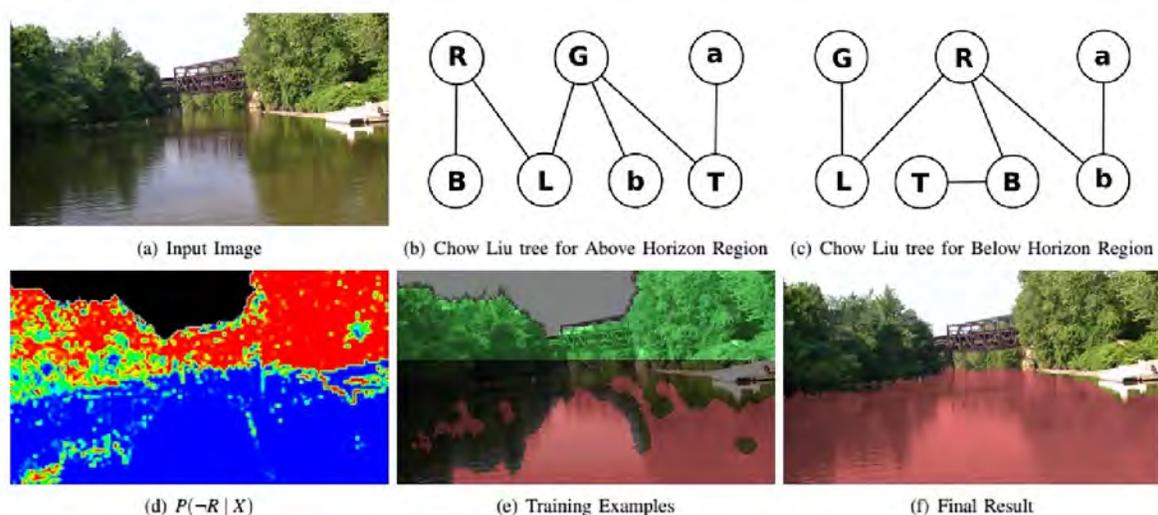


Figura 3.4 – Sequência do algoritmo auto supervisionado para extração das características de classificação e detecção de regiões aquáticas [1].

Os trabalhos relatados nesta seção usam descritores robustos com a finalidade de combinar características e construir classificadores de aprendizagem de imagem [1, 20, 16]. A partir de um conjunto de dados e um treinamento, é possível distinguir superfícies aquáticas. Esta união forma um bom classificador e o uso de um método de treinamento com base nessas características apresenta vantagem, pelo pressuposto que ambientes naturais sofrem variações prévias, e com isso determina uma segmentação mesmo em diferentes alterações que os ambientes naturais sofrem.

3.4 Comparação dos trabalhos apresentados

Considerando o estudo realizado, os trabalhos apresentados e pensando no tamanho do veículo aquático e seu processamento, para este trabalho, conforme descrito no capítulo 3, foi adotada uma câmera monocular e o uso de várias características baseadas na coloração da água para conseguir o máximo de informações para a segmentação da imagem [36, 1, 47, 23]. São usados blocos de pixel para a extração das características, criando um conjunto de critérios treinados com um método supervisionado, que aumenta o desempenho e quantifica a porcentagem de acerto em métodos usando classificação [1]. A combinação de características é quantificada pela porcentagem de pixels corretamente marcados em média sobre o conjunto de dados de avaliação [36]. A partir de um método de aprendizagem que deriva de um modelo off-line treinado apenas uma vez, e com o método supervisionado, foi construído um classificador que recebe um conjunto de pares de valores (água e não água) rotulados para representar um ambiente [16]. Após um treinamento, então, é possível com o valor de cada entrada fazer uma verificação seguindo critérios definidos (usando características) e gerar uma saída correspondente para cada conjunto dos pares de entrada [20].

A tabela 3.1 apresenta de forma resumida alguns trabalhos abordados neste capítulo, com informações sobre a finalidade de cada um, técnicas utilizadas para a segmentação da superfície aquática, resultados obtidos e uma comparação com o método proposto. São apresentados resumidamente alguns pontos que poderiam ser melhorados com o método proposto ou as combinações de algumas técnicas que poderiam alcançar melhores resultados, de modo a obter contribuições para os trabalhos realizados.

Uma das principais diferenças do trabalho desenvolvido em comparação com os apresentados neste capítulo é a construção de um classificador baseado em um conjunto de características. Estas características foram compactadas por um método de redução de dimensionalidade treinado para reconhecer e identificar as propriedades ópticas que a superfície aquática turva apresenta.

Tabela 3.1 – Resumo das informações apresentadas por alguns trabalhos relacionados.

Trabalho	Finalidade	Técnica utilizada	Resultados	Comparação com o método proposto
Gong et al. [16]	Algoritmo para traçar uma linha de borda entre a água e a terra com base na cor e homogeneidade da textura ou outras características visuais apresentada pela superfície aquática.	Limiar adaptativo, medição de energia, simetria com base na cor, textura, e reflexões, rede neural.	O tempo de processamento é cerca de 2 segundos em uma imagem de tamanho 320 x 240. Os erros de treinamento e teste alcançados pela rede <i>backpropagation</i> treinados são de 2,53% e 3,8%, respectivamente.	Apresentou bons resultados, mas a criação de dois algoritmos para cada discriminação da superfície aquática, com a extração das características e treinamento da rede, tornou o processamento muito demorado. A classificação de 11 imagens foi feita em 27 segundos. Usar apenas um treinamento com um conjunto de características torna o classificador mais rápido e menos complexo para processamento.
Ranking et al. [37]	Detectar corpos de água para um veículo autônomo de superfície, considerando a variação de cor.	Análise de cor em imagem RGB, saturação e brilho HSV. Filtro variância sobre as imagens em tons de cinza e variação de intensidade.	A partir de distâncias de 50, 35 e 37 metros, respectivamente, as taxas de detecção de verdadeiro positivo e falso negativo foram (95,76%, 96,71%, 98,77%) e (0,45%, 0,60%, 0,62%), respectivamente. O tempo médio para classificar cada imagem de tamanho 512x384 foi de 128,76 milissegundos.	Usa várias técnicas entre espaços de cores para estimar a contribuição total do coeficiente de reflexão a partir de reflexões de superfície e a combinação de todos os outros fatores. A combinação destas características com um método de treinamento aumentaria a detecção de verdadeiros positivos.
Heidarsson et al. [20]	Técnica para gerar mapa de obstáculo aquático com classificação de imagens aéreas. Com a combinação das imagens e dados de um sonar, coletados de um veículo, foi possível modelar um mapa com três classes para a navegação de veículos aquáticos: obstáculo, transifório/suspeito obstáculo e não-obstáculo.	Histograma de componentes de cor. Combinação de cor e textura, com espaço de cor RGB, HSV e CIELab. Para cada canal dos modelos de cores foram calculados os valores da intensidade, média energia e entropia com desvio padrão dos valores de intensidade.	Os experimentos foram realizados em duas áreas, e as imagens usadas foram do Google maps e Bing. O cálculo das taxas de erro de classificação, foi comparado com os resultados de imagens segmentadas manualmente, obstáculos e detectado respectivamente: Área 1 imagens Bing 6,6% 5,7% e imagens Google 8,3% 7,8%; Área 2 Imagens Bing 6,2% 6,1% e imagens Google 8,6% 7,8%.	Foram usadas imagens de satélites, sendo que algumas podem conter distorções, ou o próprio rio sofrer alterações. Aplicar a ideia em um veículo aquático, coletando imagens e classificando, contém uma confiabilidade maior. A combinação de textura e cor contribuiu significativamente para a identificação dos obstáculos, mas com o incremento de espaços de cor e cálculo de textura como variância, aumentaria a combinação de características.
Santana et al. [40]	Propor um modelo para detecção de água em sequências de vídeos para operação de navegação de veículos autônomos de superfície em ambiente natural. Ao pesquisar as imagens de entrada o modelo assume que a superfície da água apresenta tipicamente uma textura dinâmica e é capaz de filtrar o fundo e qualquer objeto dinâmico presente na cena.	Técnica de segmentação e binarização baseada na extração de imagens de entrada das regiões perceptíveis de possuir uma textura dinâmica, calculado a entropia como forma de buscar um melhor resultado, aplicado o fluxo óptico sobre as imagens de entrada e para melhorar o resultado especificamente quando a superfície aquática apresentar ondas, é aplicado um método de propagação de rótulo.	Os resultados apresentados foram qualitativos sobre 12 vídeos do banco de dados DynTex, e quantitativo para 25 <i>frames</i> classificados sobre cada vídeo, sendo que a métrica de desempenho foi 0,95% taxa verdadeiro positivo contra uma taxa de 0,1% de falso positivo.	Pela taxa de verdadeiro positivo e falso positivo o modelo apresenta ser bastante robusto e eficiente na identificação da superfície aquática, entretanto foi possível constatar na conclusão que o modelo é prejudicado pelo movimento da câmera, em que é explicitamente mencionado no trabalho que os rastreadores de fluxo óptico no processo de pré identificação, tiveram alguns valores de falsos positivos anulando em sequências de imagens em que houve grande movimentação da câmera.
Achar et al. [1]	Busca desenvolver um método auto supervisionado aplicado ao contexto das várias influências que sofrem ambientes aquáticos.	A combinação dos espaços de cores RGB e HSV foi usada em conjunto e individualmente em várias combinações para treinar os classificadores. Descritores de textura foram usados em conjuntos de Laws' masks [26], com uma janela 3x3 para cada Laws' Masks.	Tempo de execução de processamento em imagens de tamanho 640x360: <i>Feature Extraction</i> (cor) 0.49s; <i>Feature Extraction</i> (textura) 1.08s; Aparência <i>Modeling</i> 0.41s; Formação Classificador 0.20s; Detecção final 0.14s, total 2.32s. Quando combinados recursos de cor e textura apresentaram o menor erro 4.61% após avaliação do classificador supervisionado.	Apresentou resultados muitos bons com a combinação de características e o uso de um conjunto de descritores, tornando o método bastante robusto. O uso de dois classificadores, um acima da linha do horizonte e outro abaixo, para a detecção da superfície apresentou boa classificação, mas tornou o modelo complexo e custoso, assim tratar só casos exclusivos pode alcançar resultados similares.

4. DESCRIÇÃO DO MÉTODO

A navegação em ambientes *outdoors* é mais complexa do que em ambientes internos e controlados. Em ambientes aquáticos os desafios são maiores pela diversidade e variedade de elementos, como folhas, detritos, grama, vegetação, entre outros. Fatores climáticos e fatores físicos como intensidade luminosa, sombras e reflexão também são elementos que influenciam diretamente nas cores e texturas apresentadas pelas superfícies aquáticas [23]. Neste capítulo é descrita tanto a metodologia aplicada, como a estruturação e implementação do método proposto para a identificação de superfície aquática turva.

4.0.1 Metodologia

O desenvolvimento do método proposto, que é baseado em visão computacional, considera as propriedades ópticas apresentadas pela cor da água. Portanto, foi elaborado um classificador para discriminar duas classes distintas: regiões navegáveis e regiões não navegáveis. O classificador tem a finalidade de criar um mapa de navegabilidade que indica a superfície da região navegável, a qual o veículo aquático pode percorrer com confiança. Com esta informação, um algoritmo de navegação executa tomada de decisões baseado no mapa de navegação construído. Um detalhamento das etapas que fazem parte deste método é apresentado a seguir.

O processo de identificação e classificação começa com a coleta de imagens ou sequência de imagens obtidas através de uma câmera de vídeo, acoplada em um veículo aquático, por exemplo. Estas imagens correspondem às superfícies que se deseja navegar. Cada imagem coletada é subdividida em blocos, e o tamanho de cada bloco é definido de maneira a assegurar um bom desempenho computacional. Para o conjunto de pixels de cada bloco são extraídas e calculadas várias características, de forma a gerar atributos baseados em medidas estatísticas e espaços de cor. Após extraídas todas as características, é aplicada uma transformação linear para mudança de coordenadas, de forma a representá-las no espaço rotacionado obtido via *Principal Component Analysis* (PCA). Essa transformação possibilita uma diminuição da dimensionalidade dos dados de entrada, pois apenas os componentes principais do novo espaço são considerados. Estes dados transformados em uma nova base são passados para o classificador treinado e modelado com redes neurais artificiais do tipo *Multi Layer Perceptron*. A rede é executada e o classificador gera um mapa de navegabilidade que tem a finalidade de distinguir as regiões navegáveis de não navegáveis. Após esse processo, um algoritmo de navegação baseado em uma máquina de estados finitos interpreta o mapa de navegabilidade para tomar ações, como "virar a direita", "seguir em frente" ou "parar", determinando para cada imagem qual ação

deve ser executada. Ao completar esta sequência de passos para uma imagem, uma próxima imagem é coletada do ambiente e o processo recomeça. A Figura 4.1 apresenta o diagrama que contém todas as etapas do processo de identificação apresentado.

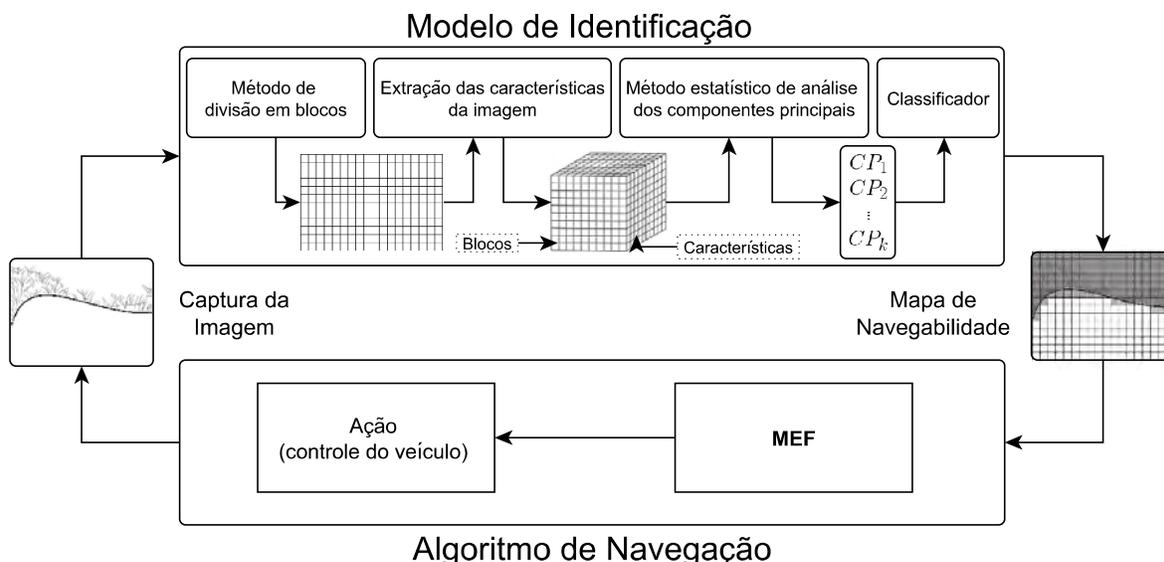


Figura 4.1 – Diagrama com as etapas do processo de identificação da superfície aquática.

As seções a seguir apresentam uma descrição de cada etapa desta metodologia com o objetivo de detalhar cada parte da criação e desenvolvimento do método para a identificação da superfície aquática turva. Sendo assim, o texto está organizado da seguinte forma: A seção 4.1, apresenta o método de subdivisão em blocos e a transformação das imagens para processamento; a seção 4.2, descreve o método de extração das características que usa medidas estatísticas; a seção 4.3 detalha como foi reduzida a dimensão das características, preservando o máximo das informações geradas; a seção 4.4, apresenta como foi construído o classificador baseado em redes neurais, bem como seu treinamento e avaliação; e a última seção descreve como foi desenvolvido o algoritmo de navegação, que toma ações com base no mapa de classificação.

4.1 Divisão em blocos

A primeira etapa do método, consiste em coletar imagens obtidas de uma câmera acoplada em um veículo aquático. A Figura 4.2(a), apresenta um exemplo de uma imagem de entrada. Estas imagens são subdivididas em um conjunto de subimagens denominadas blocos, para serem processadas posteriormente como uma única unidade. Assim, cada imagem P de tamanho $height \times width$, denominado $(H_g \times W_d)$, que for usada para a seleção das características, será decomposta em blocos B de tamanho $(r \times s)$ divisíveis de

($Hg \times Wd$). Na etapa de divisão dos blocos sempre foi considerada uma divisão exata da imagem. A Figura 4.2(b) apresenta a imagem subdividida, na qual cada quadrado é um bloco.

Portanto, cada imagem P é subdividida em pequenos blocos quadrados B , com lado $r = s$, como mostra a Figura 4.2(b). Os valores de s e r devem ser divisores tanto de Wd como de Hg . Supondo que se tenha uma imagem P de tamanho ($Hg \times Wd$), o valor do elemento $P(hg, wd)$, corresponde ao valor do pixel na linha hg e coluna wd da imagem. Assim, cada bloco denominado $B(r \times s)$ contém todos os pixels $P(hg, wd)$ que o constituem.

Com a subdivisão da imagem P em blocos, muitas características, ou propriedades, de cada bloco podem ser processadas, cada uma de forma independente e com uma função específica. Estas características serão utilizadas para posterior análise e para distinguir uma região navegável de uma região não navegável. Foi adotado o método de divisão em blocos com o propósito de aumentar o desempenho, avaliando assim, as informações de um conjunto de pixels diretamente conectados e com o objetivo de reduzir a quantidade de informações geradas em cada imagem, o que permite um processamento mais rápido.

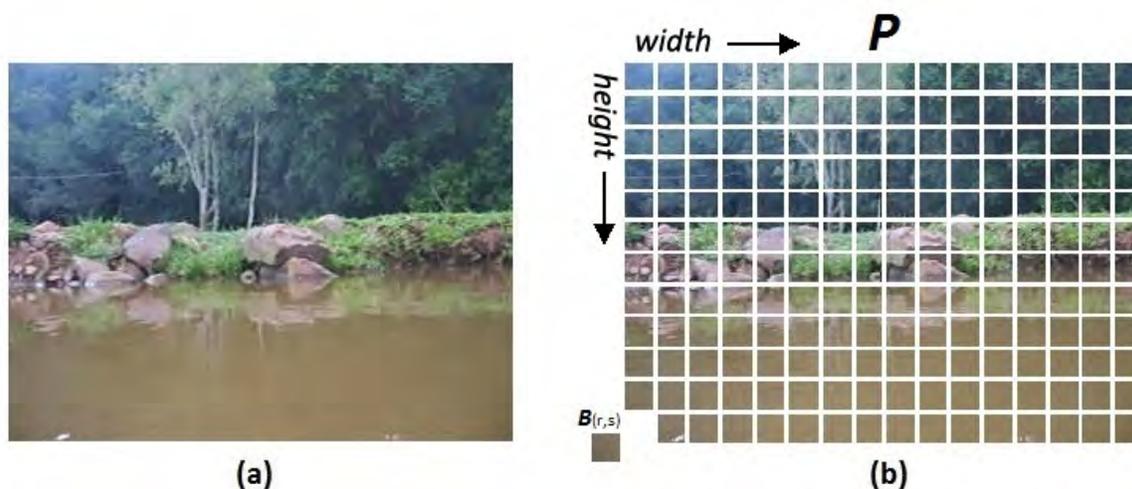


Figura 4.2 – Imagem de entrada (a) e imagem P subdividida em blocos de tamanho $B(r, s)$ (b).

4.2 Extração das características das imagens

Na etapa de extração das características das imagens são aplicadas medidas estatísticas para cada conjunto de pixels constituintes de um bloco. Portanto, para cada bloco, atributos de textura são calculados individualmente para cada espaço de cor. Sendo assim, a extração das características se refere à forma de geração destes atributos. Cada bloco pode ser representado por um atributo ou característica, e o conjunto destes atributos é

utilizado como entrada para o classificador de aprendizado de imagem, para determinar a qual classe cada bloco pertence, ou seja, se é uma região navegável ou não navegável.

4.2.1 Cálculo dos atributos de textura

Para cada espaço de cor apresentado na seção 2.1, foram calculadas medidas estatísticas que serão utilizadas para a geração dos atributos para o classificador. Estas medidas estatísticas correspondem a propriedades que representam uma análise de similaridade ou uma tendência de se encontrar um pixel ou um conjunto de pixels em uma imagem.

Dado um bloco B de tamanho $(r \times s)$ é possível realizar vários cálculos estatísticos para caracterizá-lo. Para isso, é calculado o histograma normalizado da intensidade de cada canal c . Cada histograma será denotado por H_c , onde $c \in \{ Red, Green, Blue, Hue, Saturation, Value, Y(luminance), U(chrominance) \}$. O valor do elemento h_i a partir do histograma H_c é dado por:

$$h_{ci} = \frac{n_i}{n}, \quad (4.1)$$

onde n_i é o número de pixel com intensidade i em cada canal c de um determinado bloco B de imagem, $n = r \times s$, e $i \in [0, M]$, sendo M o valor máximo de intensidade do canal de cor (por exemplo, $M = 255$ considerando uma profundidade de cor 8 bits por pixel).

Dados os 8 histogramas normalizados em H_c , são calculadas as estatísticas de média, entropia, variância e energia. As definições e equações estão descritas a seguir.

Média:

A média dos valores pode ser obtida pela equação 4.2:

$$v_c = \sum_{i=0}^{M-1} i \cdot h_{ci}. \quad (4.2)$$

Entropia:

O conceito de entropia foi introduzido por Shannon [44], para medir a quantidade de informação transferida por um canal ou gerada por uma fonte. A entropia é definida matematicamente na equação 4.3:

$$E_c = - \sum_{i=0}^{M-1} h_{ci} \log_2 h_{ci}. \quad (4.3)$$

Variância:

Na teoria da probabilidade e na estatística, variância é um conceito muito conhecido. A variância de uma variável aleatória ou processo estocástico é também definida como uma medida da sua dispersão à média, que pode ser calculada e obtida pela equação 4.4,

$$\sigma_c^2 = \sum_{i=0}^{M-1} (i - v_c)^2 \cdot h_{ci}, \quad (4.4)$$

onde v é a média da intensidade dos pixels do bloco c .

Energia:

O valor de energia do bloco c , mede a presença de valores elevados em relação a outros valores, e pode ser definida pela equação 4.5:

$$\varepsilon_c = \sum_{i=0}^{M-1} (h_{ci})^2. \quad (4.5)$$

4.2.2 Considerações

Existem vários outros atributos que podem ser calculados a partir de uma imagem além das medidas estatísticas e características apresentadas. Atributos que podem ser calculados a partir de um único pixel, de um conjunto de pixel ou de uma imagem inteira. Este trabalho combina um conjunto de atributos que é utilizado para a realização da classificação baseada em alguns trabalhos [36, 1, 47, 23]. Este conjunto de atributos inclui os espaços de cor RGB, HSV e YUV extraídos de cada conjunto de pixels (bloco) e as medidas estatísticas calculadas para cada espaço, que são a média, entropia, variância e energia. Assim, chega-se a um conjunto de 32 atributos que são utilizados para a construção do classificador de aprendizado de imagem. A tabela contendo todos os atributos calculados nesta etapa se encontra no apêndice A.

4.3 Seleção das características para gerar o classificador

Uma vez extraídas todas as 32 características (Tabela A.1), dos conjuntos de blocos que compõem uma imagem, é necessário escolher quais delas são mais relevantes ou que combinação de características utilizar para identificar com maior acurácia uma superfície navegável, dada a grande variabilidade de aparências que podem ter em ambientes externos. Visando um bom desempenho na qualidade da classificação, todas as características e atributos extraídos são importantes ou contribuem para determinar a qualidade na geração de um classificador que possa ser robusto e eficiente. Entretanto, o uso de atributos com alta dimensionalidade tem grande influência em um método de aprendizado, podendo impedir a convergência ou causar instabilidade [19].

Uma metodologia matemática alternativa adotada para diminuir a dimensão dos dados de entrada é *Principal Component Analysis* (PCA) [58], que é amplamente utilizada em situações em que seja necessário uma redução da dimensionalidade de variáveis, sem consequente perda na quantidade de informações disponíveis [24]. Por exemplo, no trabalho de Haykin [18], foi utilizado PCA como uma etapa de pré-processamento dos dados utilizados, reduzindo a complexidade no projeto do classificador baseado em redes neurais. Segundo Haykin, um número reduzido de variáveis retém a maior parte da informação intrínseca, assim a redução da dimensionalidade se dá pela eliminação das combinações lineares entre as variáveis de entrada da rede, restando apenas aquelas que expressarem a maior parte da variabilidade dos dados originais. Deste modo para reduzir a quantidade de dados de entrada para o método de aprendizado de máquina foi aplicado o método PCA, descrito a seguir.

4.3.1 *Principal Component Analysis*

Principal Component Analysis é considerado um método estatístico multivariado linear que permite a redução da dimensionalidade de grandes conjuntos de dados de entrada. Resumidamente, a análise de componentes principais consiste essencialmente em reescrever as coordenadas das amostras em outro sistema de coordenadas mais conveniente para a análise dos dados. É um método que substitui as variáveis originais α por um número menor de variáveis derivadas β , em que os componentes principais são combinações lineares das variáveis originais [52].

Em outras palavras, as n variáveis originais geram, através de suas combinações lineares, k componentes principais, cuja principal característica, além da ortogonalidade, é que são obtidos em ordem decrescente de variância. Ou seja, o primeiro eixo do componente principal tem a direção da combinação linear das variáveis de entrada com maior

variabilidade, seguido pelo segundo eixo de componente principal que detém menos variabilidade de dados que o primeiro eixo, mas máxima informação ortogonalmente ao primeiro eixo. O terceiro eixo detém menos variabilidade de dados que o segundo eixo, mas máxima informação ortogonalmente ao segundo eixo e assim por diante.

A utilização do PCA como uma etapa de pré processamento para um método de aprendizado de máquina pode, pela diminuição da correlação entre as novas variáveis derivadas, acelerar a convergência de um método de aprendizado como de retropropagação [18]. PCA, além de preservar o máximo possível das informações das variáveis iniciais, é uma metodologia de compreensão de dados [24], já que algumas variáveis de entrada muitas vezes possuem informações redundantes ou não apresentam alguma contribuição significativa para a construção e modelagem de um classificador.

Normalização e padronização dos dados

Uma etapa inicial para a redução de dimensionalidade é a preparação dos valores das variáveis de entrada. No projeto de construção de um método de aprendizado de máquina, a normalização tem a finalidade de adaptar os dados de entrada à faixa dinâmica das funções de ativação. Funções como sigmoid logística ou a tangente hiperbólica tem como saída apenas valores entre 0 e 1, casos como redes neurais em que a tangente hiperbólica (na camada intermediária) possuem valores muito altos, podem saturar a função de ativação prejudicando a convergência da rede neural [27]. O processo de normalização é dado pela equação 4.6:

$$\hat{f}_j = \frac{f_j - f_{min}}{f_{max} - f_{min}}, \quad (4.6)$$

na qual \hat{f}_j é o valor normalizado da características j , sendo k o total de características diferentes. De modo que, para cada característica, f_j é o valor original da característica, $j = 1, 2, \dots, k$; f_{min} é o valor mínimo da característica j e f_{max} é o valor máximo da característica j , considerando todos os blocos de treinamento.

Após a normalização dos dados, é aplicado um recurso estatístico de padronização. A padronização consiste em calcular para cada característica j a média μ_j e o desvio padrão σ_j . Cada característica extraída \hat{f}_j foi normalizada de acordo com a equação 4.7

$$z_j = \frac{\hat{f}_j - \mu_j}{\sigma_j}. \quad (4.7)$$

Cada z_j representa uma característica com média μ_j e desvio padrão σ_j , e sobre cada valor \hat{f}_j , a característica z_j recebe o valor padronizado. Cada nova característica z_j terá

média zero e desvio padrão igual a 1. Assim, a padronização corresponde ao deslocamento do centro (dado pela média) de um conjunto de dados para a origem do sistema cartesiano.

Após os dados serem normalizados e padronizados, eles são organizados em uma matriz $X_B X_j$, na qual B representa o número de blocos da imagem e j representa a quantidade de características extraídas.

O PCA nada mais é que o cálculo dos autovalores e correspondentes autovetores de uma matriz de variâncias - covariâncias entre variáveis. Assim, o primeiro passo é encontrar a matriz de covariância Cov_Z , dada pela equação 4.8:

$$Cov_Z = Z.Z^T \quad \equiv \quad Cov_Z = \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1p} \\ c_{21} & c_{22} & \cdots & c_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ c_{p1} & c_{p2} & \cdots & c_{pp} \end{pmatrix}, \quad (4.8)$$

na qual, Z^T é a transposta de Z . Os elementos da diagonal principal desta matriz se referem às variâncias (λ) das colunas (variáveis independentes). Já os elementos fora da diagonal principal representam a covariância entre as variáveis. A matriz de covariância é simétrica, de modo que o elemento c_{ii} na diagonal é a variância de cada X_i e o termo fora da diagonal c_{ij} é a covariância entre as variáveis $X_i X_j$. Quando a covariância é nula significa que as variáveis aleatórias são descorrelacionadas.

Em seguida são calculados os autovalores (λ) e autovetores (ν) da matriz Cov_Z . Existem p autovalores, que podem ser zero, mas, não é possível existir autovalores negativos para uma matriz covariância. Deste modo os autovalores estão ordenados como $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$, então λ_i corresponde ao i -ésimo componente principal, como mostra a equação 4.9,

$$\lambda_i = a_{i1} X_1 + a_{i2} X_2 + \cdots + a_{ip} X_p, \quad (4.9)$$

em que, λ_i e as constantes $a_{i1}, a_{i2}, \dots, a_{ip}$ são os elementos do autovetor correspondente. Para cada autovalor (λ_i), existe um autovetor (ν_i). Os autovetores são normalizados, isto é, a soma dos quadrados dos coeficientes é igual a 1, e ainda são ortogonais entre si, como mostra a equação 4.10:

$$\sum_{i=1}^p a_{ip}^2 = 1. \quad (4.10)$$

No PCA os autovalores da matriz de covariância correspondem à variância das variáveis transformadas. Assim, se um autovetor possui autovalor grande, significa que este fica em uma direção em que há uma grande variância dos padrões. Este é um fato importante para a classificação, pois, é mais fácil diferenciar padrões usando uma base em que seus vetores não são correlacionados e que apontam para a direção da maior variância dos dados [31].

Os autovetores representam a direção, ou seja, a contribuição com a qual cada um dos eixos originais entra na composição dos novos eixos dos componentes principais. Os autovalores, por sua vez, representam a quantidade de variância original descrita pelos respectivos autovetores [48].

Contribuição de cada componente principal

Como o objetivo fundamental do PCA é a redução da dimensão das variáveis de um conjunto de dados, é preciso saber a contribuição de cada componente principal. A porcentagem de cada componente é calculada pela equação 4.11, que divide a variância de um λ_i pela variância total, acumulando em PC_i a porcentagem explicada pelo componente principal λ_i ,

$$PC_i = \frac{\lambda_i}{\sum_{i=0}^p \lambda_i} . 100 \quad (4.11)$$

Existem vários critérios para encontrar o número k componentes principais a ser substituído pelas variáveis originais, equilibrando a redução da dimensão dos dados com a necessidade de explicar o máximo da proporção importante da variância total. No PCA a redução de dimensão se dá com a eliminação dos autovetores com autovalores mais baixos, é conhecido que informações também podem ser perdidas na eliminação dessas variáveis e que o tamanho dessa redução pode ser dada pela porcentagem das variáveis eliminadas, menos a soma dos valores da equação 4.11.

Segundo Ian [24] é difícil determinar a dimensão ideal de componentes principais, mas um critério de seleção simples é o de minimizar a complexidade considerando um limite para se manter fiel ao problema. Ou seja, buscar um limite máximo de componentes principais que expresse uma alta quantidade dos dados originais e que não impacte nas questões de desempenho. Um critério bastante conceitual é por avaliações do gráfico *scree plot*, que é a plotagem dos autovalores. Por exemplo, na figura 4.3 o ponto de corte foi o \hat{d} , definido por análise dos valores inferiores a um limite \hat{r} , valores que estão abaixo deste intervalo serão eliminados. Ou por diferença entre pares de valores, em que essa diferença for menor que 10-20%.

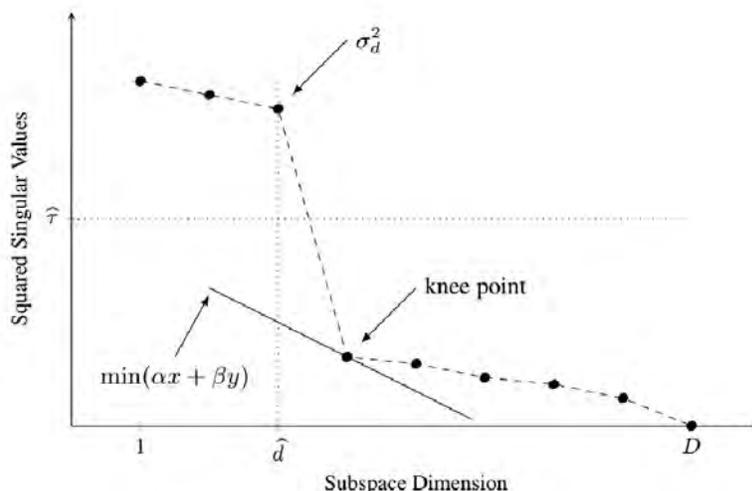


Figura 4.3 – Imagem de um gráfico *scree plot* que relaciona o número de componentes D de cada eixo com o valor próprio \hat{r} que lhe está associado [24] .

Escores

Os escores são as projeções das amostras na direção dos componentes principais. Suas coordenadas são obtidas a partir do produto da matriz de dados pela matriz de autovetores [31]. Seu cálculo é obtido pela equação 4.12, para os X_n dados da matriz.

$$\begin{aligned}
 SC_{11} &= \nu_{11}X_{11} + \nu_{12}X_{12} + \cdots + \nu_{1p}X_{1p} \\
 SC_{21} &= \nu_{11}X_{21} + \nu_{12}X_{22} + \cdots + \nu_{1p}X_{2p} \\
 &\vdots \\
 SC_{n1} &= \nu_{11}X_{n1} + \nu_{12}X_{n2} + \cdots + \nu_{1p}X_{np} .
 \end{aligned} \tag{4.12}$$

Por fim, sobre cada eixo dos componentes é realizada uma diagonalização. A matriz P é empregada para criar a matriz de mudança de base em Cov_z obtendo uma matriz diagonal D de autovalores de Cov_z , pela equação 4.13

$$D = P^{-1} C_z P . \tag{4.13}$$

A matriz D apresenta elementos iguais aos autovalores na diagonal principal, reescrevendo os valores de cada componente principal em uma nova base. Ao extrair as características de um bloco no processo de classificação esses dados são aplicados sobre a matriz de mudança de base para fazer a "tradução" dos novos dados sobre os componentes já calculados (seção 4.4.4). Ou seja, muda-se a base dos dados e calcula-se os componentes de um vetor arbitrário correspondente a uma nova base [30, 31].

4.3.2 Considerações

Com o objetivo de aumentar a velocidade de convergência e generalidade no reconhecimento de padrões em processos de treinamento, foi utilizado o método de PCA. Um método de redução de dimensionalidade que auxilia na elaboração de hipóteses gerais a partir dos dados coletados, contrastando com testes de padrões nas principais variabilidade presentes sobre o eixo de cada componente. Vale ressaltar que os componentes principais selecionados permitem reduzir a dimensão do conjunto de dados, e não o número de variáveis originais, já que os componentes são combinações lineares de todas as variáveis originais.

De forma análoga, na etapa de classificação, é utilizada só a matriz de mudança de base, pois é a ligação entre os novos dados a serem processados com a transformação linear dos componentes por meio de seus valores em uma base.

4.4 Classificador

A finalidade do classificador é identificar em uma imagem que região corresponde a uma superfície navegável, determinando para cada bloco (ou conjunto de pixels) a que classe ele pertence. Para cada imagem a ser classificada, são definidas duas classes de acordo com o resultado do classificador: navegável (blocos brancos) e não navegável (blocos pretos). Assim, o conjunto de todas as características e atributos extraídos com base nos componentes principais é utilizado pelo classificador para realizar esta classificação.

O classificador pode ser definido da seguinte maneira: Seja B um bloco de uma imagem a ser classificada então $CP = \{ CP_1, CP_2, \dots, CP_k \}$ é o conjunto de todos os componentes principais para determinar a classificação. Assim, o classificador recebe um conjunto de dados baseados nos componentes principais como entrada, e retorna a qual classe este conjunto pertence. A Figura 4.4 representa esse processo.

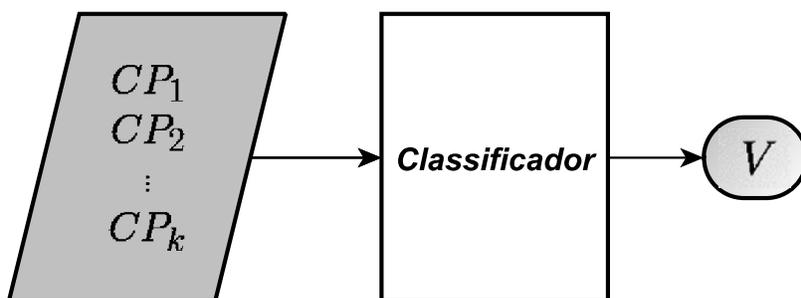


Figura 4.4 – Classificador destinado a receber os valores referentes aos CP .

Para a construção do classificador, foi adotado um método de aprendizado de máquina conhecido por Redes Neurais Artificiais (RNAs), do tipo *Multi Layer Perceptron*. Alguns conceitos básicos sobre redes neurais, são abordados na fundamentação teórica, bem como a arquitetura utilizada na implementação do classificador, descrevendo as formas gerais de treinamento.

4.4.1 Avaliação do classificador

A avaliação de uma rede neural normalmente é feita pela complexidade de como é construída e pela forma como os seus parâmetros são modelados, ou seja, quantas entradas, camadas intermediárias ou saídas possui a arquitetura da rede neural. Segundo Yu Lean et al. [61] o cálculo da complexidade dos modelos de redes neurais é feito pela equação 4.14:

$$CX(n) = n_h(n_i + 1) + n_o(n_h + 1), \quad (4.14)$$

em que $CX(n)$, é a complexidade do modelo, n_i é o número de entradas, n_h é o número de neurônios na camada intermediária ou oculta, e n_o é o número de neurônios na camada de saída. Através da equação 4.14, é possível perceber que a complexidade de um modelo de rede neural pode ser reduzida através de um tratamento adequado de preparação de dados, como, por exemplo, a redução de entradas [61].

Ao modelar uma rede neural, devem ser feitas análises para examinar a eficiência e a adequação do modelo construído. Deste modo, avaliações devem ser conduzidas para determinar quando uma rede neural está concluída ou afirmar sua confiabilidade. Existem inúmeras formas de se avaliar uma RNA. Uma análise bastante conhecida, por exemplo, é o EQM, “Erro Quadrático Médio”, em que ao treinar uma RNA sua finalidade é que seja igual a zero ou que chegue a um valor mínimo aceitável. Segundo Haykin [19, 46], nem sempre um EQM baixo implica em uma boa generalização, uma vez que o EQM não informa quantas entradas foram mal classificadas. Assim, se o erro for alto em alguns padrões na entrada de treinamento, ele será distribuído proporcionalmente entre os resultados da saída.

Outra forma de avaliar o desempenho de uma RNA é estimar uma porcentagem de acerto para cada padrão que obteve uma classificação correta, métrica de desempenho conhecida como *Hit Rate*. O uso desse método implica em conhecer muito bem o conjunto de dados utilizados para definir uma boa precisão. O uso de uma precisão alta demais pode considerar muitos resultados como errados, fazendo a taxa de acerto da RNA ficar muito baixa, e uma precisão baixa demais pode gerar uma taxa de acerto elevada, mas um resultado final insatisfatório [46].

Neste trabalho, com o objetivo de planejar uma avaliação da convergência da RNA na fase de treinamento, foi utilizado o método desenvolvido por Shinzato et al. [45]. Este método atribui um peso para o erro de classificação para determinado neurônio, computando uma pontuação para S . Neste método, um maior peso é atribuído a uma saída de RNA com erro de 0,1 do que com uma saída com erro de 0,2. Através desta pontuação ponderada, há uma tendência de "recompensa" para neurônios com menos erros grandes ou vários erros pequenos, e de "punição" para os demais. A equação 4.15 mostra como é calculada essa pontuação.

$$S = \frac{\frac{1}{N \cdot p(0)} \left(\sum_{i=0}^{hmax} h(i) \cdot p(i) \right) + 1}{2.0}, \quad (4.15)$$

onde N é o número de classes avaliadas, $h(i)$ é o número de classificações com erro variando de $\frac{i}{hmax}$ a $\frac{i+1}{hmax}$, e $hmax$ é o número de intervalos a ser considerado para a discretização do processo de contagem de erro. O valor de $hmax$ determina a precisão para a interpretação da saída da RNA. Em outras palavras, por exemplo, se $hmax = 10$, então a saída com um valor real que varia de 0 a 1 será dividida em 10 classes de erros: um intervalo de erros para valores entre 0,0 e 0,1, outro intervalo de erros entre 0,1 e 0,2, e assim por diante. Para simplificar o manuseio do método de avaliação S , os pesos variam de -1 a 1, portanto, S varia de 0 a 1. Por exemplo, se $S = 1$ significa que todas as saídas da RNAs da avaliação estão corretas. Portanto, a etapa de formação decorre até pontuação converge para 1 ou algum valor mais próximo.

4.4.2 Processo de treinamento do classificador

Após a definição da forma de avaliação de uma RNA, um passo comum e bastante importante, assim como qualquer método de aprendizado de máquina, é realizar o processo de treinamento. Inicialmente, o usuário deve coletar e armazenar um conjunto de imagens para permitir o treinamento dos dados que serão realizados pelas RNAs. O conjunto de imagens coletadas para a extração dos dados deve ser bastante amplo, com diferentes vistas de um cenário, e até mesmo com mais de um cenário. Com as imagens de treinamento deve-se realizar o processo de divisão das duas regiões disjuntas da imagem, ou seja, manualmente o usuário deve marcar as regiões navegáveis e não navegáveis. Para este fim, uma ferramenta interativa foi construída em que através de interações com o mouse é realizada a distinção da região de acordo com a presença de água ou não. A figura 4.5 ilustra como é realizado o processo de anotação.

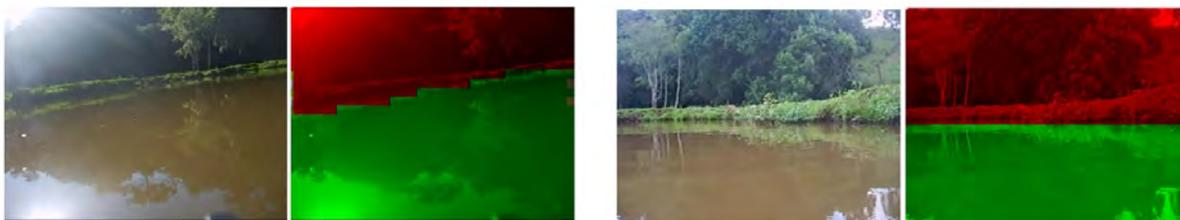


Figura 4.5 – Exemplo de duas imagens segmentadas manualmente para uso no treinamento. As regiões navegáveis são pintadas de verde e as não navegáveis de vermelho.

No processo de anotação é realizada a atribuição dos pesos para cada bloco pertencente ao conjunto das imagens de treinamento. Blocos sinalizados com vermelho possuem peso "0" e blocos sinalizados com verde possuem peso "1". Com as imagens devidamente sinalizadas, são extraídas as características e os escores dos componentes principais de cada bloco, seguindo a metodologia definida nas seções 4.2 e 4.3. Ao salvar a anotação realizada é gerada uma matriz contendo esses dados e uma matriz de peso para cada bloco. Estas informações serão utilizadas posteriormente para o treinamento da RNA com o algoritmo de *Resilient Propagation*. O treinamento é instanciado com um ciclo de 5 mil ou até a RNA alcançar uma pontuação S aceitável, acima de 95%. Estes parâmetros determinam a qualidade do resultado e buscam influenciar o máximo possível o treinamento para alcançar o resultado ótimo.

4.4.3 Modelagem final do classificador

Após apresentados os conceitos e metodologias aplicadas para avaliar e treinar a RNA, é descrito como foi construída a arquitetura, uma vez que a estrutura do classificador se baseia diretamente na quantidade de variáveis de entrada da RNA. Por fim é apresentado o processo final do classificador construído de forma a mostrar o resultado gerado, em que é possível identificar duas classes de superfície aquáticas com o mapa de navegabilidade.

A partir do cálculo dos componentes principais (seção 4.3), obtidos das coleções dos blocos das imagens, é possível estruturar a quantidade de camadas de entrada da RNA. A camada de entrada se baseia na quantidade de componentes principais, uma vez que os dados de entrada são o conjunto de escores dos componentes principais das imagens de treinamento. Ao realizar todo o processo de definição da quantidade de neurônios na camada de entrada, é definida a quantidade de neurônios na camada intermediária. Alguns autores [14, 62, 21] dizem que é sempre uma incógnita afirmar a quantidade de neurônios em uma camada intermediária, pois se for muito grande a rede estará memorizando padrões e com isso perdendo a capacidade de generalização, por outro lado se a camada for muito pequena a rede levará muito mais iterações para atingir a precisão dese-

jada. Empiricamente, com base na literatura, observou-se que para cada 3 neurônios na primeira camada era preciso 1 neurônio na camada oculta, ou para cada saída pode se ter o dobro na camada oculta. Com base nisto foi definida a equação 4.16, onde hid define a quantidade de neurônios na camada intermediária, sendo CP_k a quantidade de total de componentes encontrados

$$hid = \frac{CP_k}{2} . \quad (4.16)$$

Para a camada de saída foi definido 1 neurônio pois como é preciso classificar apenas duas classes de regiões, 1 neurônio é suficiente para suprir a necessidade da classificação. Como apresentado no início da seção 4.4.2 pela equação 4.14, a complexidade de uma RNA pode ser calculada pela sua arquitetura. Assim, quanto mais simples sua modelagem, maior será a velocidade de treinamento e conseqüentemente melhor será sua generalização [46]. A figura 4.6 ilustra o modelo final para o classificador construído.

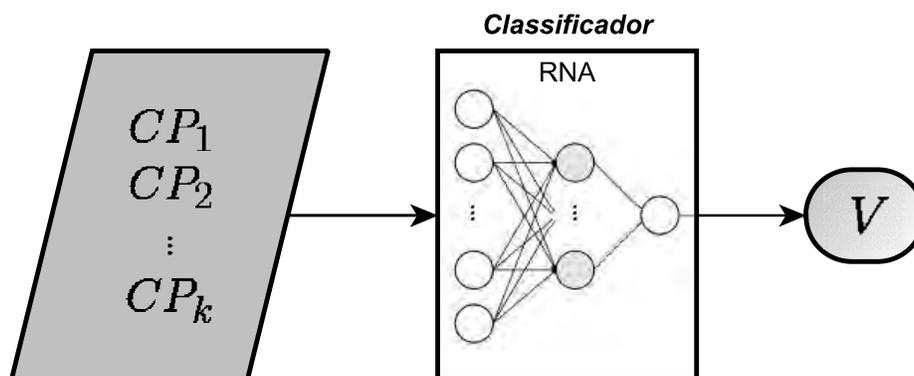


Figura 4.6 – Modelagem final do classificador. As entradas são componentes da imagem a ser classificada, e a saída é o valor calculado pela classificação.

Depois que o classificador é modelado e a RNA é treinada, ele é instanciado e executado, e para cada imagem a ser classificada é obtido um mapa visual de navegabilidade. A figura 4.7(a), apresenta algumas imagens a serem classificadas, e o resultado gerado pelo classificador, que é a conversão da imagem de entrada em tons de cinza, é mostrado na figura 4.7(b).

Cada imagem a ser classificada é subdividida em blocos, sendo que cada bloco recebe uma classificação $V \in [0, 1]$. Assim, quanto mais próxima de 1 ($V \geq 0.5$) for a saída da RNA, que é classificação de um bloco, mais branco é atribuído à sua cor e, conseqüentemente, maior é a certeza de que seja uma região navegável. Isso significa que o conjunto das características deste bloco é considerado mais navegável. De forma oposta, quanto

mais próxima de 0 for a classificação de um bloco ($V < 0.5$), mais preto é seu bloco e maior é a certeza de que é uma região não navegável.

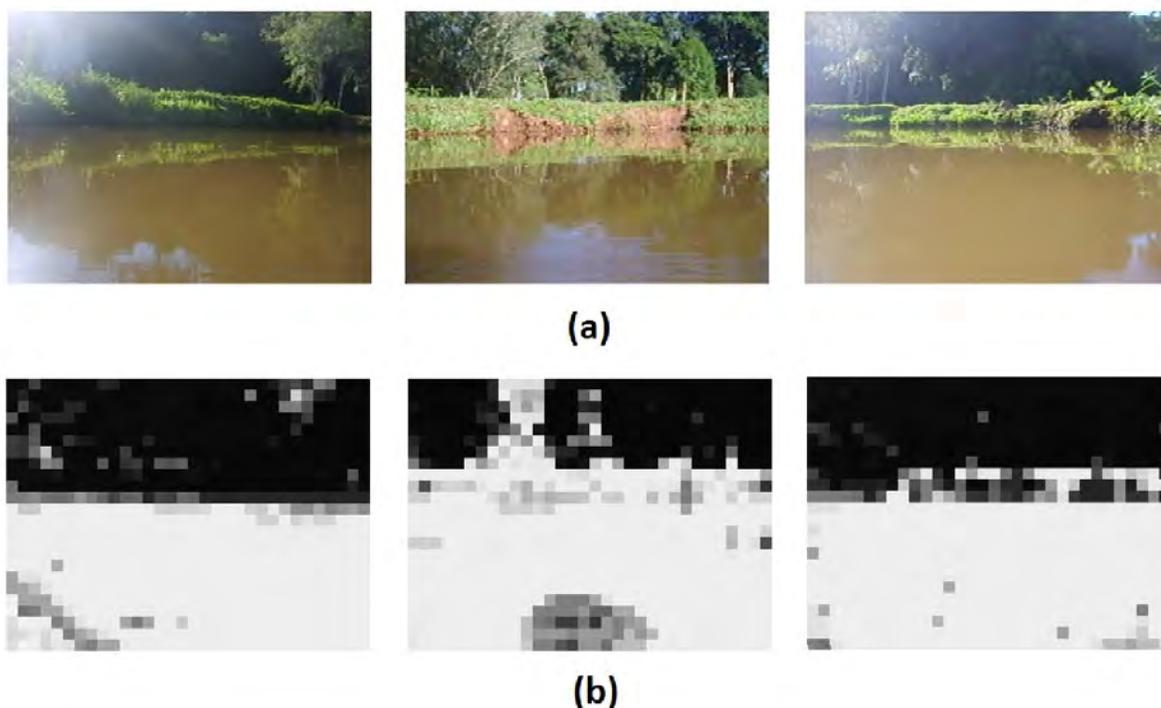


Figura 4.7 – Imagens a serem classificadas (a), obtidas pelo veículo aquático da figura 4.1(a), e mapas de navegabilidade destas imagens classificadas (b), nos quais quanto mais branco maior é a certeza de região navegável.

4.4.4 Classificando novas imagens

Após a implementação do método descrito nas seções anteriores, o passo seguinte foi incorporá-lo em um veículo aquático, como protótipo, para permitir a realização de testes e avaliação. Uma descrição deste protótipo é apresentada na seção 5.4. Assim, quando o veículo é colocado na superfície aquática, novas imagens são adquiridas pela câmera acoplada. Cada nova imagem é subdividida em blocos e para cada bloco são processadas as medidas estatísticas para os 8 espaços de cor, conforme descrito nas seções 4.1 e 4.2. Estas características são, então, normalizadas e padronizadas, gerando os novos valores dos escores.

Os escores dos novos blocos de imagem são projetados nas coordenadas de espaço do PCA. Para este fim, foi calculada uma multiplicação de matrizes sobre a matriz de mudança de base para a alteração da base das características extraídas:

$$SC = D.Z \text{ ,} \quad (4.17)$$

sendo D a matriz de autovetores ordenados e obtidos pelo PCA (equação 4.13), e Z o vetor de características normalizadas e padronizadas extraídas de cada bloco.

Os escores SC dos k componentes principais, correspondente de cada bloco, são, então, submetidas para as RNAs. Cada bloco é classificado gerando uma saída, uma matriz na qual cada elemento corresponde a um bloco de imagem classificado, gerando o mapa de navegabilidade. Este mapa guiará a tomada de decisões sobre a direção que o veículo deve seguir. A próxima seção descreve o desenvolvimento do algoritmo de navegação e processo de tomada de decisão que foi implementado.

4.5 Navegação

Após a construção do método de identificação, foi desenvolvido um algoritmo de navegação para interpretar o mapa de navegabilidade gerado, a fim de definir ações sobre o veículo para que possa navegar por um ambiente. Considerando os vários desafios que a navegação em ambientes reais apresentam, seria possível chegar a decisões ou construir alternativas para executar ações de problemas complexos e de natureza dinâmica com tomada de decisão [54]. Desta forma, foi proposto um algoritmo para navegação baseado no conceito de Máquina de Estados Finitos (MEF). Este algoritmo é matematicamente definido como um quintupla, 4.18:

$$Mc = \{Q, I, f, q_0, F\} \text{ ,} \quad (4.18)$$

onde Q representa o conjuntos de estados; I representa as entradas para a MEF; f a função de transição de estados; q_0 o estado inicial; F o conjunto de estados de parada da MEF. As tomadas de decisões são baseadas no mapa de navegabilidade, o qual o algoritmo avalia, toma a decisão dentro dos estados em que se encontra (por exemplo, "vira a esquerda" ou "siga em frente"), e envia esta informação para que o veículo acione seu hardware e execute a ação sobre o ambiente. Todo o processo desde a identificação e navegação pode ser visualizado na figura 4.1.

O processo de análise do mapa de navegação pelo algoritmo para a tomada de decisões é baseado em áreas pré definidas, como mostra a Figura 4.8(a), na qual cada área consiste em uma entrada para a MEF. Para cada imagem classificada são definidas as regiões $SP1$, $SP2$, $SP3$ e $SP4$, que são delimitadas pelas linhas $L1$, $L2$ e $L3$. Com as áreas definidas, é realizada uma busca para determinar quais áreas estão com maior

probabilidade para tomar a ação de navegação. As linhas $L2$ e $L3$ da Figura 4.8(a), são definidas, respectivamente, através das seguintes equações:

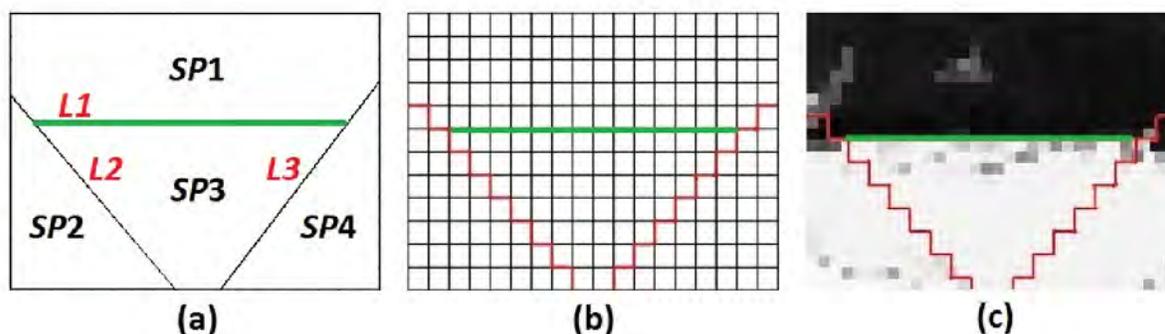


Figura 4.8 – Definição das áreas para as tomadas de decisões (a); Busca das classificações baseada nas áreas pré definidas (b); Combinação entre a áreas pré definidas com o mapa de navegabilidade (c).

$$\begin{aligned} y_2 &= \frac{1}{3}nr + x_2, \\ y_3 &= \frac{1}{3}nr + nc - x_3, \end{aligned} \quad (4.19)$$

onde nr é o número de linhas de blocos e nc é o número de colunas de blocos, do mapa de navegabilidade gerado. A definição das regiões $SP2$ e $SP4$ é feita através dos índices de y_2 e y_3 , que representam as linha da matriz de blocos do mapa de navegabilidade, e x_2 e x_3 , que correspondem aos índices das colunas do mapa de navegabilidade. Assim os valores y_2 e x_2 são, respectivamente, a linha e a coluna de $L2$, e y_3 e x_3 são a linha e a coluna de $L3$. A origem está no canto superior esquerdo do mapa navegação e y é orientado de cima para baixo. A figura 4.8(b), mostra a busca sobre o mapa de navegação de forma a quantificar cada área para a tomada de decisão. A linha $L1$ é identificada automaticamente pela classificação da primeira RNA, sendo definida pela linha superior dos blocos que têm pelo menos 25% da classificação como "navegável"(ver seção 5.3.2 para maior detalhes).

Por exemplo, se a maioria dos blocos pertencentes a área $SP2$ possuir a classificação $V \geq 0.5$, ou seja, mais próxima a 1, e a maioria dos blocos das áreas $SP3$ e $SP4$ possuir uma classificação mais próxima a 0 ($V < 0.5$), vira-se à esquerda. De forma contrária, se a maioria dos blocos pertencentes a área $SP4$ possuir a classificação $V \geq 0.5$, vira-se à direita. Todas as áreas são comparadas, e a ação sempre consiste em ir na direção da área que tiver o maior número de blocos navegáveis.

Uma vez que as áreas são delimitadas, uma MEF define se o veículo deve permanecer na mesma direção, virar à esquerda, virar à direita ou parar. A Figura 4.9, apresenta o diagrama de sequências da MEF para as tomadas de ações de acordo com análise realizada sobre o mapa de navegabilidade. Existem outros estados que poderiam ser desenvolvidos para o controle de navegação, mas o escopo deste trabalho se limita aos quatro

comandos definidos em Q , que são "para frente", "virar a direita", "virar a esquerda" e "parar".

Portanto, inicialmente é contado o número de blocos classificados como "navegável" em cada região. O processo de decisão pode ser resumido da seguinte maneira:

- Se a linha do horizonte, $L1$, for inferior a $\frac{2}{3}.nr$, o veículo deve parar. Isto ocorre principalmente quando há poucos (ou nenhum) blocos navegáveis na frente do veículo;
- Se o número de blocos classificados como "navegável" em $SP2$ é maior do que em $SP3$ e $SP4$, o veículo deve virar à esquerda;
- Se o número de blocos classificados como "navegável" em $SP3$ é maior do que em $SP2$ e $SP4$, o veículo deve seguir em frente;
- Se o número de blocos classificados como "navegável" em $SP4$ é maior do que em $SP2$ e $SP3$, o veículo deve virar à direita;
- Se o número de blocos classificados como "navegável" em $SP2$, $SP3$ e $SP4$ for igual, a prioridade é sempre tomar a ação de seguir em frente.

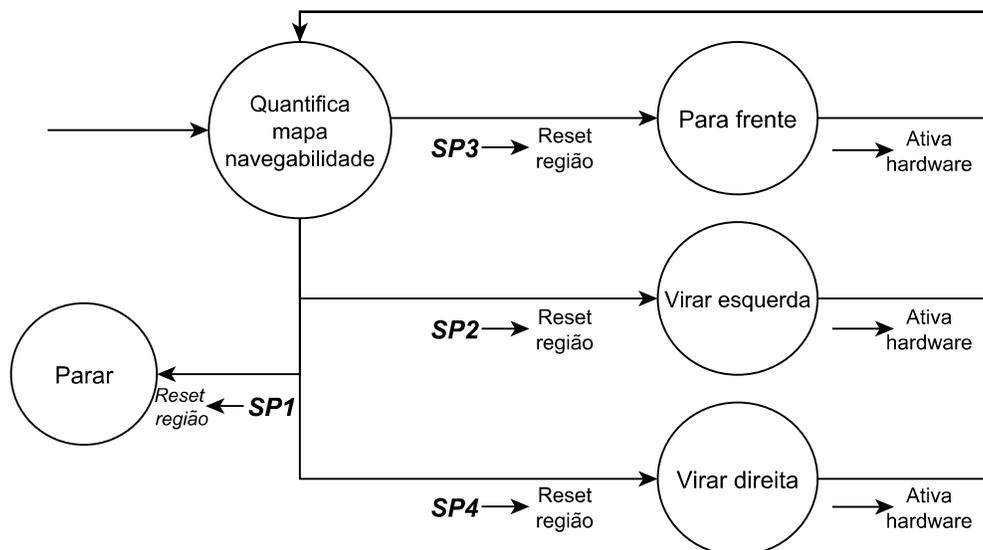


Figura 4.9 – Diagrama de sequências das ações da MEF.

4.5.1 Consideração final

A abordagem baseada em tomada de decisões é uma simples prova de conceito, desenvolvida para garantir desempenho rápido quando incorporada no veículo aquático.

Foi usada uma MEF porque é simples de descrever uma sequência de estado, facilitando a mudança de um estado para outro, ou a definição de novos estados, considerando diferentes contextos para cada imagem de entrada. A adoção do uso de uma MEF é uma proposta para demonstrar a navegação com base no mapa de navegabilidade gerado, uma vez que todo o método foi embarcado a um veículo aquático.

Outros métodos poderiam ser aplicados para interpretação do mapa de navegabilidade, visando a extração de mais informações para o gerenciamento das ações a serem tomadas pelos veículos em determinados estados que poderiam surgir no trajeto em navegação aquática aberta.

Para complementar a metodologia apresentada até o momento, o capítulo 5 descreve uma série de experimentos isolados, com o objetivo de mostrar o desempenho e a precisão do método proposto. Métricas de desempenho e avaliações sobre os resultados e os testes, validam e ressaltam a descrição do método, bem como o entendimento do trabalho apresentado.

5. TESTE E RESULTADOS

Visando analisar e validar o método proposto, foram realizados vários experimentos com o objetivo de coletar dados e informações para fundamentar e complementar a metodologia explicada, alcançando, assim, os resultados esperados para este trabalho. Esta coleta de dados e informações foi feita em ambientes reais, em diferentes condições, para se aproximar ao máximo das condições de navegação que o veículo aquático poderá encontrar. Mais especificamente, foram coletados vídeos gravados com uma câmera acoplada em um veículo aquático controlado remotamente em diferentes ambientes. Os trajetos foram percorridos em duas lagoas após um forte período de chuva, para conseguir obter uma característica na superfície aquática com indicie de turbidez elevada.

A Figura 5.1(a) apresenta o veículo de superfície aquática utilizado para coletar as imagens. Este veículo é um barco estilo lancha controlado remotamente e que foi adaptado com uma estrutura para ser acoplada uma câmera para a coleta do vídeo. A câmera utilizada foi uma Samsung L100, configurada a uma resolução de (320 x 240) pixels com 30 FPS. A Figura 5.1(b) ilustra uma imagem coletada pela câmera que foi acoplada ao veículo. Esta imagem apresenta a diversidade de um ambiente real, como a vegetação presente na encosta e a reflexão do céu sobre a água.



Figura 5.1 – Barco utilizado para a coleta dos dados (a) e um exemplo de imagem coletada do ponto de vista do veículo aquático (b).

As ferramentas utilizadas para a construção e desenvolvimento do método proposto foram a linguagem de programação C, com auxílio das bibliotecas OpenCV e FANN (*Fast Artificial Neural Network Library*). A biblioteca OpenCV foi utilizada para trabalhar principalmente no processamento, controle e exibição das imagens [3]. A biblioteca FANN implementa redes neurais artificiais de multicamadas na linguagem C [12], sendo, portanto, usada na criação, gerenciamento e execução da rede neural. As demais funções e rotinas como o PCA, foram desenvolvidas na linguagem C [33] (seção 4.2). Para os experimentos,

foi definido que o tamanho do bloco seria de $r = s = 10$ pixels (seção 4.1), uma vez que as imagens possuem tamanho de 320 x240, o que significa que cada imagem coletada possui a quantidade de 768 blocos.

De forma a separar os testes e os resultados, esse capítulo está dividido da seguinte forma. A seção 5.1 apresenta o conjunto inicial de *frames* utilizados para treinamento e as informações necessárias para a extração dos dados e o aprendizado do ambiente, que visam gerar o conhecimento necessário para diferenciação e distinção na classificação do ambiente a ser analisado. A seção 5.2, apresenta como foi modelada e construída a rede neural, bem como os parâmetros analisados em sua fase de treinamento, almejando o máximo desempenho para a classificação. A seção 5.3 apresenta os resultados obtidos com a classificação de algumas sequências de imagens em vários cenários para demonstrar o desempenho do classificador construído e melhorias sobre o próprio classificador com variações dos dados de entrada com o intuito de analisar a adaptação do classificador em diferentes condições que podem surgir em ambientes reais. Por fim, a seção 5.4 descreve como foi construído e embarcado o método proposto, bem como o hardware utilizado para a construção do protótipo de um veículo aquático, onde foi implementado o algoritmo de navegação. Avaliações e testes do algoritmos de navegação são apresentados na tomada de decisão baseado em MEF de forma a validar a aplicação de um veículo autônomo desenvolvido para navegar em superfícies aquáticas turvas.

5.1 Definição dos parâmetros iniciais

Nesta seção é apresentado o conjunto de imagens usadas para extrair todas as características para gerar os dados necessários para o treinamento do classificador. A primeira etapa consistiu na coleta de um vídeo de treinamento. Para isto, foi gravado um vídeo treinamento de aproximadamente 10 minutos usando o veículo apresentado na Figura 5.1(a). Deste vídeo foram extraídos 15 *frames* aleatórios que foram salvos como o conjunto de *frames* de treinamento, que é apresentado no apêndice B.

Com o conjunto de treinamento definido, cada *frame* passou pelo processo de divisão em blocos. Em seguida, foi realizada uma segmentação manual dos blocos sobre cada *frame* de entrada, com a finalidade de atribuir pesos. De acordo com o processo de treinamento supervisionado do classificador, é necessário fazer esta atribuição de pesos sobre cada bloco dos *frames* de treinamento. Sendo assim, foi realizado o processo de anotação, que consistiu em pintar os blocos para identificar as duas classes de região: navegável e não navegável. Nesta etapa, cada bloco foi segmentado e recebeu um peso, como detalhado na seção 4.4.2. Portanto, para cada bloco segmentado com cor verde foi atribuído peso 1, sinalizando uma região navegável, e para cada bloco segmentado com cor vermelha foi atribuído peso 0, sinalizando uma região não navegável. Esse processo de

anotação de pesos serve tanto para a extração das características detalhado na seção 4.2, quanto para o treinamento da RNA, uma vez que se consegue distinguir as duas classes de regiões presentes em cada *frame* e extrair toda informação, gerando deste modo os dados necessários para a criação dos atributos e os valores dos pesos dos bloco para a etapa de treinamento. Como comentado na seção 4.3.1, estes dados estão em escalas de padrões diferentes, eles são normalizados e padronizados para a etapa seguinte de redução de dimensionalidade por análise de componente principal, gerando os escores dos *CPs*.

5.1.1 Redução da dimensionalidade por análise de componente principal

Após a definição da coleção dos *frames* de treinamento, atribuição de pesos e cálculo das características para cada bloco de cada *frame* com os valores normalizado e padronizado, a matriz de dados para redução de dimensão está pronta. Seguindo a metodologia apresentada nas seções 4.2 e 4.3, esta matriz de dados corresponde às informações disponíveis e ao conjunto de dados para o cálculo dos componentes principais e redução de dimensionalidade. Componentes que contêm a combinação de todos atributos necessários para o treinamento e criação do classificador, responsável pela distinção e identificação de cada classe presente nos *frames* a serem classificados pelo classificador.

O PCA é aplicado com a finalidade de reduzir a dimensão dos conjuntos de dados e definir a quantidade de neurônios de entrada da RNA, por seleção dos CP_k primeiros componentes. A figura 5.2 apresenta o gráfico *scree plot*, calculado sobre a matriz de entrada. Com o gráfico, foi possível definir a quantidade de $CP k = 6$, de acordo com a metodologia apresentada na redução de dimensionalidade por análise de gráfico *scree plot* e o critério de seleção de um limite $\hat{r} \geq 1$ de autovalores superiores a esse limite \hat{r} .

A tabela 5.1 apresenta os resultados das variâncias (autovalores), a porcentagem de variância e a porcentagem cumulativa obtida com os componentes principais, para os seis primeiros *CPs* definidos. A inclusão de mais *CPs* aumentaria a dimensionalidade sem aumentar muito a representatividade dos autovalores, uma vez que a porcentagem de variância do CP_7 e CP_8 representam 2,00% e 1,61%, respectivamente, somando uma porcentagem de variância quase equivalente ao CP_6 .

Através da tabela 5.1 é possível verificar que a maior quantidade de variância dos dados é do primeiro componente principal, CP_1 , o qual representa aproximadamente 45% da variância total dos dados originais. De acordo com Ian et al. [24], com a variância dos seis primeiros *CPs* foi possível expressar 90,73% dos dados originais. Ou seja, todas as 32 variáveis originais podem ser analisadas a partir desses seis *CPs*, o que garante uma redução das variáveis de aproximadamente 81%. Por fim, com a seleção dos principais *CPs*, o conjunto de dados com 32 variáveis, foi reduzido a seis variáveis, formando, assim, o conjunto de dados de treinamento.

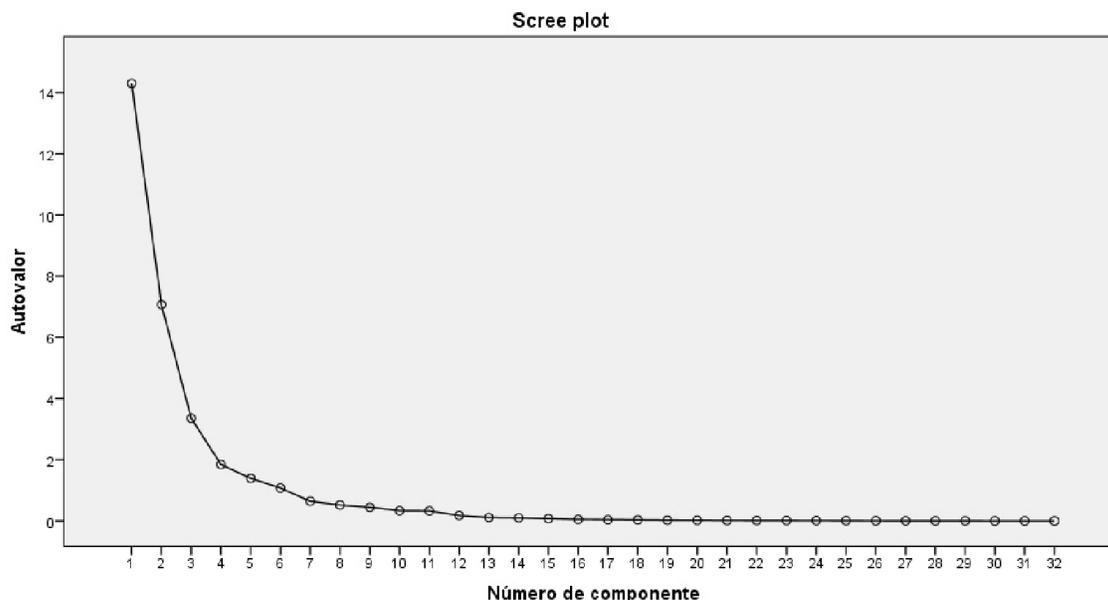


Figura 5.2 – Gráfico *scree plot* dos componentes calculados.

Tabela 5.1 – Tabela de valores do total de variância explicada por componente.

Componentes	Valores próprios iniciais		
	Autovalores	% de variância	% cumulativa
CP_1	14,304	44,699	44,669
CP_2	7,069	22,092	66,791
CP_3	3,353	10,477	77,269
CP_4	1,844	5,761	83,030
CP_5	1,393	4,354	87,384
CP_6	1,073	3,352	90,736
⋮	⋮	⋮	⋮

Para entender como cada variável impacta em cada componente, uma forma de visualização é utilizar uma plotagem do gráfico de escores dos componentes principais, que é o conjunto de dados de treinamento. O gráfico da Figura 5.3 apresenta a distribuição dos escores projetados no subespaço definido para os três primeiros componentes CP_1 , CP_2 e CP_3 .

Com o gráfico da Figura 5.3, é possível observar a distribuição e projeção dos escores rotulados no subespaço definido pelos três primeiros componentes. Pode-se observar nesta figura uma boa separação dos blocos classificados como navegável (vermelho) dos não-navegáveis (azul), e uma distinção e separação não-linear. Essa é a razão pela qual nós empregamos uma rede neural artificial como um classificador, pois devido a este fato as coordenadas dos dados neste subespaço são usadas como entradas para a formação e treinamento da rede neural. A próxima seção aborda como os CPs caracterizam a rede neural, bem como cada entrada para o treinamento e classificação da rede neural.

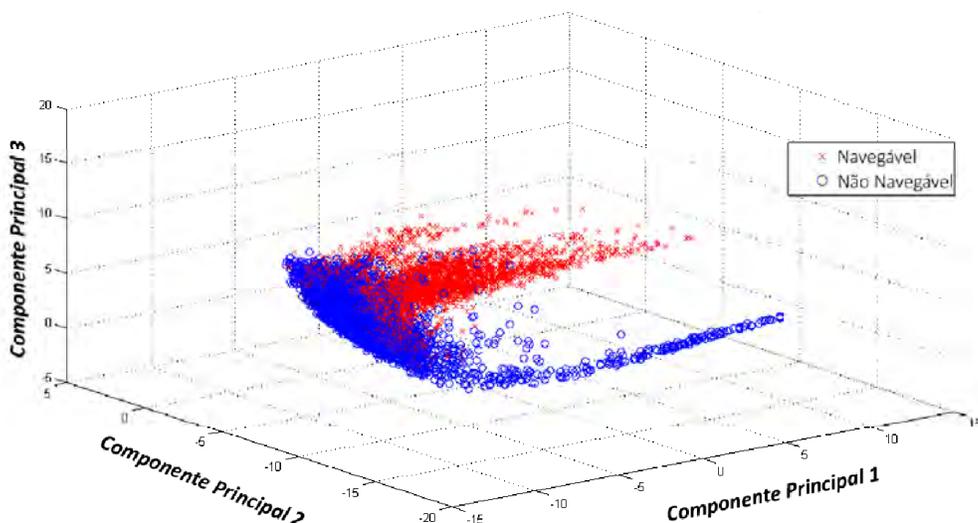


Figura 5.3 – Características padronizadas a partir de blocos de treinamento projetado no subespaço definido pelos três componentes principais do PCA.

5.2 Rede neural e avaliações

Com a definição dos componentes principais, e após o cálculo do conjunto de dados de treinamento, é construída a rede neural. A figura 5.4 apresenta a modelagem final da rede neural, na qual os seis neurônios correspondentes aos escores dos seis primeiros componentes principais definiram a camada de entrada, a camada intermediária com três neurônios e a saída com um neurônio. Com a RNA arquitetada foram realizadas análises e avaliações com o objetivo de coletar informações sobre o treinamento e o grau de certeza que a RNA possui sobre os padrões determinados, conforme definido na seção 4.4.1. O objetivo da avaliação é buscar a importância dos padrões classificados erroneamente, para os resultados em que a diferença esperada e a obtida é pequena.

Alguns parâmetros já foram definidos para a realização dos testes sobre a RNA arquitetada. Assim, a taxa de porcentagem de acerto variou de 0,1 a 0,3. Os demais parâmetros como função de ativação não foram modificados ao longo do treinamento, sendo que para a função de ativação na camada intermediária e para a camada de saída foi utilizada a função sigmoideal.

Vale ressaltar que a função de ativação sigmoideal aplicada nos neurônios da camada intermediária é útil para introduzir a não linearidade nas RNAs. Sem linearidade, as camadas intermediárias não tornam as RNAs *multi perceptron*, deixando uma RNA simples *perceptron*, e a função linear aplicada na camada de saída é mais direcionada e apropriada para variáveis contínuas, diferente deste caso [31].

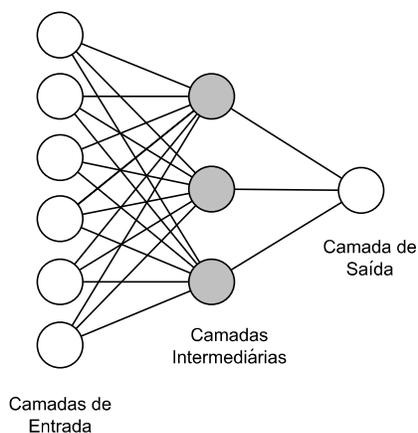


Figura 5.4 – Modelagem final da rede neural artificial.

5.2.1 Treinamento

Com os pesos e os *bias* inicializados, a RNA está pronta para o treinamento. Por se tratar de uma rede *feedforward*, foi utilizado o algoritmo de treinamento *resilient propagation*, em que os pesos sinápticos da rede são ajustados de acordo com a propagação do erro de sinal com o conjunto dos dados em treinamento. Em outras palavras, para superar as desvantagens da gradação descendente e adaptar os pesos de tal forma, que o erro quadrático da rede seja minimizado, o algoritmo *resilient propagation* realiza uma adaptação local de atualização de pesos de acordo com o comportamento da função erro. Ao contrário de outras técnicas adaptativas, o efeito do processo de adaptação do algoritmo *resilient propagation* não sofre muito pela influência imprevisível do tamanho dos derivativos, mas depende somente do comportamento temporal de seu sinal, elevando seu processo de adaptação. Por se tratar de um treinamento adaptativo é considerado um algoritmo avançado, eficiente e transparente, que alcança bons resultados para muitos problemas [8].

Como comentado na seção 4.4.1, foi utilizada uma função de pontuação com base no método de avaliação S , em que cada neurônio é recompensado quando menor for seu erro no treinamento. De forma justa, a inicialização do treinamento foi utilizando a função de randomização dos pesos de todos os neurônios, na faixa de -0.1 a 0.1 de acordo com o vetor de pontuação para contar a frequência de erro para cada padrão de classificação. E para cada configuração utilizada foram executados um total de 5 vezes o treinamento.

Quanto ao tempo de treinamento, foram adotados dois parâmetros como critérios de indicação de ponto de encerramento. O primeiro foi o número máximo de ciclos, sendo definido um limite de 5.000 ciclos. É muito comum ocorrer a generalização em determinados instantes no treinamento, causando o *overfitting* ou super ajustamento, que é quando a RNA começa a degenerar e aprender padrões incorretos e sua precisão não é a representação

da realidade [49]. De forma a contornar esse problema foi adotada uma estratégia de comparação do método de avaliação S , na qual, considerando o seu ciclo, cada recomeço é comparado à saída do método de avaliação S anterior. Deste modo, calcula-se a nova saída de S que determina a precisão para interpretação da saída da RNA, ou seja, quando a rede começar a incorporar o ruído presente nos dados, o que causa degradação na sua capacidade de generalização, é a pontuação S de resultado ótimo.

Como o critério de avaliação é dado pela pontuação S , que relaciona as informações de erro médio, é possível utilizá-lo como métrica do grau de precisão que a RNA possui sobre determinados padrões. O objetivo do método de avaliação S é dar maior importância aos padrões classificados erroneamente quando a diferença entre o resultado esperado e o obtido é muito pequena. A figura 5.5(a) apresenta o gráfico de EQM e a figura 5.5(b) a saída do método de avaliação S para um ciclo de treinamento. É possível notar que há melhorias no aprendizado entre os ciclos 170 e 1075, alcançando o valor de $S = 95.52\%$. Enquanto que EQM mostra uma convergência já no ciclo 150. Essa pequena melhora mostrou um aumento na classificação de blocos não navegáveis, mostrando maior confiança e certeza no resultado.

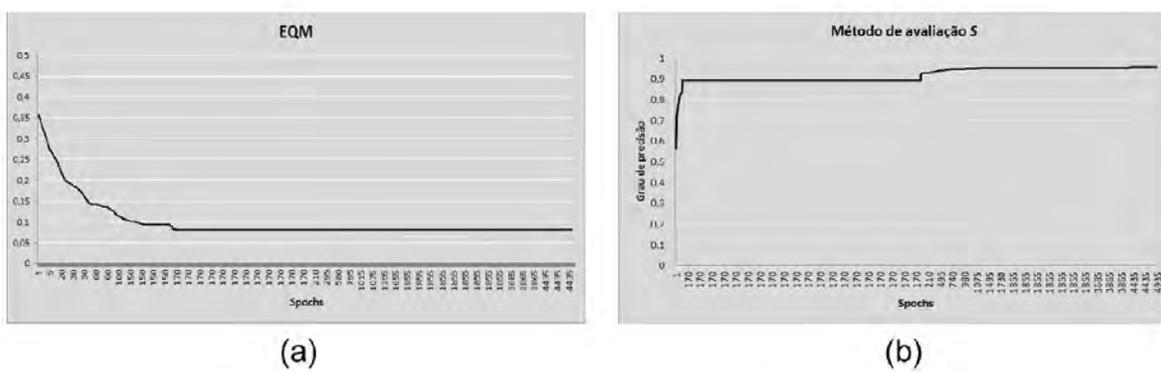


Figura 5.5 – Gráfico do valor de erro quadrático médio no treinamento da RNA (a) e valores do método de avaliação S obtidos no treinamento da RNA com 5.000 *spochns* (b).

Um exemplo desta melhoria pode ser visto na figura 5.6, na qual um *frame* foi classificado para testar a RNA. A figura 5.6(a) apresenta o resultado do *frame* classificado com a RNA cujo seu treinamento foi concluído no *spochns* 200. Já a 5.6(b), apresenta o mesmo *frame* classificado na qual a RNA teve o treinamento concluído no limite de 5.000 *spochns*. É possível observar que a RNA possui mais certeza quando há blocos mais brancos, simbolizando que a classificação é mais próxima a 1 para os blocos navegáveis. De forma similar, a RNA com o limite de *spochns* em 200 retorna cores mais escuras, simbolizando pouca certeza na classificação dos blocos.

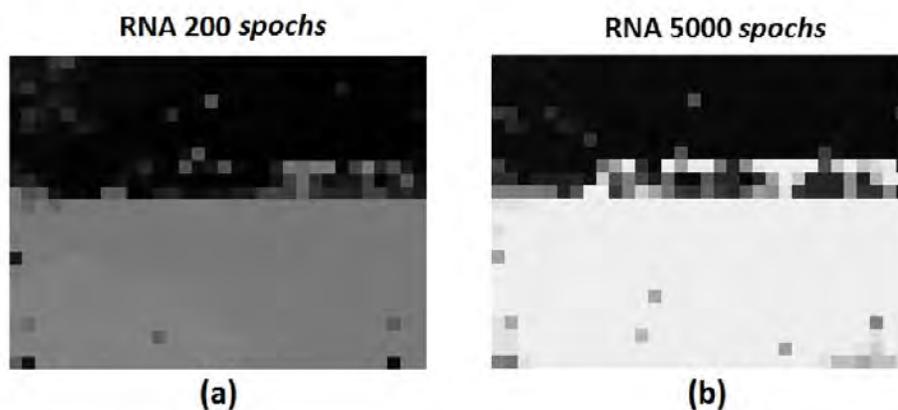


Figura 5.6 – Comparação de um *frame* classificado com diferentes treinamentos da RNA: Classificação com treinamento finalizado no *epoch* 200 (a) e no *epoch* 5.000 (b).

5.2.2 Considerações

Ao realizar os testes na fase de treinamento da RNA foi possível, perceber que o método de avaliação *S* alcança bons resultados em treinamentos de ciclos mais longos, porém, o método de EQM também possui bom desempenho em várias situações ou casos. Com a comparação entre os métodos de avaliação, foi possível constatar que o método *S* possui maior certeza em blocos navegáveis, blocos de maior interesse na navegação. Deste modo, o método de pontuação *S* foi adotado, pois força a RNA a buscar o máximo da avaliação em treinamento e maximiza a classificação e identificação das região navegáveis do ambiente.

5.3 Classificação

A fim de avaliar o método proposto, foi realizada uma fase de classificação destinada a obter uma análise, métricas e resultados do classificador construído. Para realizar esta etapa, novos vídeos foram coletados em três cenários diferentes, definidos como cenários I, II, III a fim de explorar o máximo a diversidade de ambientes. Para cada amostra foi realizada uma segmentação manual a fim de mensurar a qualidade do classificador. A segmentação manual realizada é similar ao processo de anotação (seção 4.4.2), em que para cada bloco não navegável é atribuído peso "0"vermelho e bloco navegável peso "1"verde. O resultado *V* da saída do classificador determinar qual classe cada bloco classificado pertence ($V \geq 0.5$) navegável e ($V < 0.5$) não navegáveis. Esses valores foram utilizados para fazer a análise qualitativa e quantitativa do desempenho do classificador. Com estas análises

ses busca consolidar a robustez do método desenvolvido, de forma a mensurar a qualidade e precisão do classificador.

Os vídeos de classificação são amostras de navegação de aproximadamente 5 minutos, dos vídeos foram extraídos de 15 a 25 *frames* aleatórios sobre três cenários. A extração dos *frames* foi aleatória de acordo com a quantidade de *frame* utilizados como amostra, sobre o total de *frames* do vídeo de classificação. Deste modo, são extraídos *frames* de todo o vídeo coletado, abrangendo diferentes condições do cenário e diversificando a coleção para classificação. A técnica quantitativa para desempenho do classificador foi baseada no trabalho de Fawcett *et al.* [13], chamada ROC (*Receiver Operating Characteristics*) que é uma técnica de visualização, organização e seleção de classificadores com base em seu desempenho, que têm sido muito utilizados na literatura para representar o equilíbrio entre as taxas de sucesso e as taxas de falso sucesso de classificadores.

O desempenho do classificador se dá pelos valores entre a segmentação realizada manualmente e a classificação realizada pelo classificador. As métricas de desempenho são os somatórios dos valores de: Verdadeiro positivo, que ocorre quando o resultado é positivo, assim como na realidade; Verdadeiro negativo, que ocorre quando o resultado é negativo, assim como na realidade; Falso positivo, ocorre quando o resultado é positivo, mas na realidade ele é negativo; e Falso negativo, que ocorre quando resultado é negativo, mas na realidade ele é positivo. Com estes valores é possível calcular taxas de acurácia, sensibilidade e precisão do classificador em cada cenário avaliado.

Os cálculos das métricas de desempenho são feitos através das seguintes equações:

- Acurácia:

$$\frac{\text{Verdadeiro positivo} + \text{Verdadeiro negativo}}{(\text{Verdadeiro positivo} + \text{Falso positivo}) + (\text{verdadeiro negativo} + \text{Falso negativo})} \quad (5.1)$$

- Sensibilidade:

$$\frac{\text{Verdadeiro positivo}}{\text{Verdadeiro positivo} + \text{Falso negativo}} \quad (5.2)$$

- Precisão:

$$\frac{\text{Verdadeiro positivo}}{\text{Verdadeiro positivo} + \text{Falso positivo}} \quad (5.3)$$

Assim, foi possível calcular para cada *frames* das amostras a acurácia (equação 5.1), a sensibilidade (equação 5.2) e a precisão (equação 5.3). A seguir é feita uma descrição de cada cenário avaliado e dos resultados do classificador na identificação da superfície aquática, considerando a técnica de análise de desempenho apresentada.

5.3.1 Cenário I

Inicialmente, foi utilizado o mesmo cenário do treinamento, mas com outra trajetória de navegação, para avaliar o desempenho do método. O cenário I, é um cenário rural, composto inteiramente por vegetação, árvores, pedras e grama na encosta. Ao analisar o vídeo resultante, foi possível notar a incidência de reflexo do céu na superfície aquática, alterando a propriedade óptica da superfície turva.

No apêndice C é apresentado o resultado obtido na classificação do cenário I pelo classificador. A figura C.1 do apêndice apresenta os *frames* da seguinte maneira: na coluna (a) estão os *frames* de entrada, que consistem nas amostras extraídas aleatoriamente do vídeo de classificação do cenário I; na coluna (b) são apresentados os *frames* de entrada com o resultado da segmentação manual, na qual os blocos pintados de vermelho equivalem à região não navegável e os pintados de verde à região navegável; a coluna (c) contém o resultado obtido pelo classificador para cada *frame* de entrada, no qual quanto mais preto for o bloco, maior é a certeza de uma região não navegável, e quanto mais branco, maior é a certeza de uma região navegável; na coluna (d) é realizada um sobreposição entre o *frame* de entrada e o resultado da classificação, apresentando então, a região navegável.

A tabela 5.2 apresenta em detalhes o cálculo das métricas de desempenho do classificador para cada *frame* avaliado do cenário I, com o objetivo de demonstrar de forma quantitativa a performance do classificador. A média para acurácia, sensibilidade e precisão sobre o cenário avaliado foram: 92.15 %, 87.28 %, 92.70 %, respectivamente.

Ao analisar os *frames* classificados no apêndice C.1 e os resultados de desempenho da tabela 5.2, é possível observar uma maior imprecisão nos blocos em que houve reflexo do céu sobre a água. Esta incerteza é previsível, uma vez que o classificador se baseia na combinação de características de cores e texturas para a classificação da região navegável. Por exemplo, ao extrair os atributos de uma superfície aquática que possui reflexo do céu, ao realizar a classificação com essa perturbação a rede inevitavelmente apresentará oscilação na classificação do bloco. De modo a contornar esse problema apresentado na classificação e buscar um resultado mais preciso, foi adotada a estratégia de criação de uma segunda rede neural. Esta segunda RNA é ativada pela identificação da linha de horizonte, e se destina a tratar somente estas oscilações apresentadas. Deste modo, a seguir é demonstrado como foi realizada a técnica de construção e ativação do segundo classificador.

Tabela 5.2 – Tabela de valores e métricas de desempenho do classificador para o cenário I

Frames	Verdadeiro Positivo	Verdadeiro Negativo	Falso Positivo	Falso Negativo	Acurácia	Sensibilidade	Precisão
Frame 1	358	374	10	26	95.31 %	93.23 %	97.28 %
Frame 2	365	374	10	19	96.22 %	95.05 %	97.33 %
Frame 3	370	383	1	14	98.05 %	96.35 %	99.73 %
Frame 4	362	379	5	22	96.48 %	94.27 %	98.64 %
Frame 5	361	383	1	23	96.88 %	94.01 %	99.72 %
Frame 6	348	415	1	4	99.35 %	98.86 %	99.71 %
Frame 7	334	382	2	50	93.23 %	86.98 %	99.40 %
Frame 8	282	399	17	70	88.67 %	80.11 %	94.31 %
Frame 9	212	435	13	108	84.24 %	66.25 %	94.22 %
Frame 10	148	432	80	108	75.52 %	57.81 %	64.91 %
Frame 11	168	500	44	56	86.98 %	75.00 %	79.25 %
Frame 12	305	414	49	0	93.62 %	100.00 %	86.16 %
Frame 13	332	409	23	4	96.48 %	98.81 %	93.52 %
Frame 14	308	371	29	60	88.41 %	83.70 %	91.39 %
Frame 15	318	395	15	40	92.84 %	88.83 %	95.50 %
Média do classificador:					92.15 %	87.28 %	92.70 %

5.3.2 Linha do horizonte e segunda rede neural

A detecção da linha do horizonte nos *frames* foi utilizada como forma de ativação da segunda RNA. O motivo deste experimento é contornar o problema dos blocos que sofrem com a perturbação do reflexo que causa imprecisão na classificação, uma vez que muitas imagens possuem esta adversidade. Como a região de interesse, que é superfície navegável, está abaixo da linha de horizonte, a detecção desta linha pode ser usada para a escolha de uma estratégia de análise diferente para esta região, melhorando assim, a classificação dos blocos.

A identificação da linha do horizonte $L1$ é feita durante a classificação. Para isto, cada linha de blocos no mapa de navegabilidade é analisada de cima para baixo, e a primeira em que a RNA encontrou 25% de blocos navegáveis é a $L1$. Sua altura é então calculada, determinando a sua posição no mapa de navegabilidade.

Portanto, para cada imagem P que está sendo classificada, a linha $L1$ é definida quando a contagem de blocos navegáveis B , de uma linha classificada como navegável, for maior ou igual ao valor de $\frac{1}{4}$ do total de blocos da linha. A figura 5.7, apresenta dois *frames* com o resultado obtido na identificação da linha $L1$, linha do horizonte. A figura 5.7 (a) apresenta a imagem original, enquanto figura 5.7 (b) apresenta o resultado da identificação e posição da altura da linha de horizonte no mapa de navegabilidade.

A construção da segunda RNA foi destinada a tratar a questão do reflexo que pode ocorrer nos blocos abaixo da linha do horizonte. Assim, o conjunto de treinamento desta

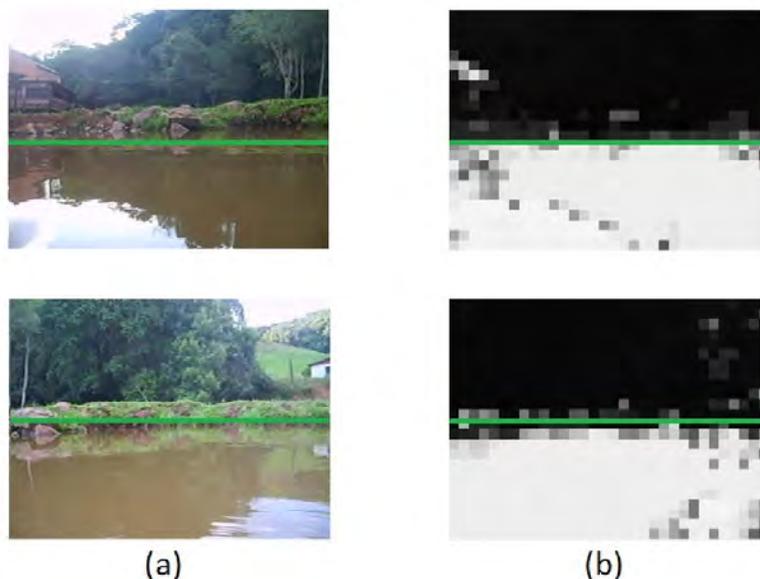


Figura 5.7 – Identificação da linha de horizonte nos *frames* do vídeo (a) e no respectivo mapa de navegabilidade (b).

segunda RNA recebeu como entrada apenas os blocos dos *frames* de treinamento que apresentavam reflexo. Utilizando o mesmo conjunto de treinamento do apêndice B, a Figura 5.8 ilustra como foi realizado o processo de anotação em dois *frames*. Seguindo o mesmo conceito apresentado na seção 4.4.2, os pesos para a segunda RNA foram limitados só a blocos com reflexo. Assim, blocos segmentados com a cor verde mantiveram o padrão de peso 1, e blocos segmentados com a cor vermelha receberam peso 0. Mas, para não haver uma contradição no processo de treinamento, blocos contendo o céu não possuem nenhum peso, de modo que blocos que não estiverem sinalizados, não entram no processo de treinamento, sendo como se não existissem.

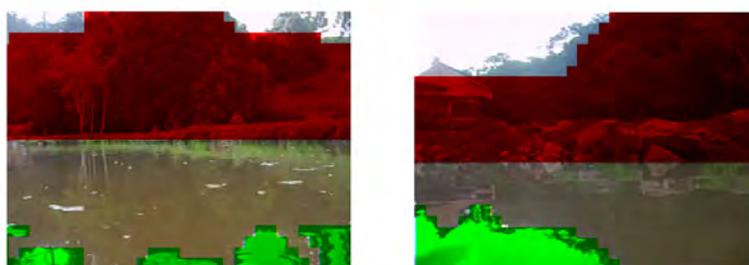


Figura 5.8 – Definição das classes navegáveis e não navegáveis para o treinamento do segundo classificador

A criação, definição e treinamento da segunda RNA seguiu a mesma metodologia apresentada na construção do primeiro classificador, descrita nas seções 4.1 a 4.4. Essa segunda RNA apresenta a seguinte topologia: quatro neurônios de entrada, dois intermediários e um de saída. Foram feitos cinco treinamentos de 5.000 ciclos e o ponto ótimo alcançado foi de $S = 96,32\%$. A Figura 5.9, apresenta o processo de utilização do segundo classificador criado para tratar possíveis reflexos apresentados pelas superfícies aquáticas.

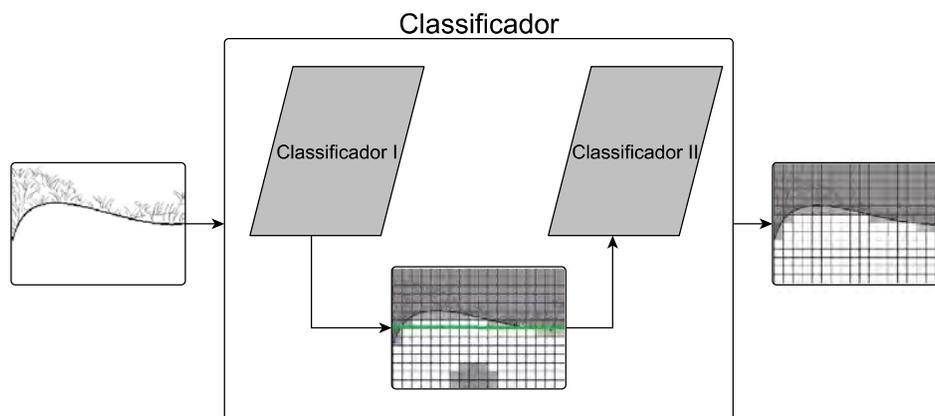


Figura 5.9 – Processo do uso de dois classificadores com base na linha de horizonte para a classificação dos *frames* de entrada.

O processo de ativação do segundo classificador se dá para cada bloco que estiver com uma classificação "possivelmente incorreta" e que estiver abaixo da linha do horizonte. Em outras palavras, se um bloco abaixo da linha $L1$ resultar em um valor $V < 0.5$ pelo primeiro classificador, este bloco é submetido ao segundo classificador para uma nova classificação. O processo de combinação de dois classificadores busca melhorar o resultado de uma possível região com reflexo sobre a superfície. Se este segundo classificador retornar um valor de $V < 0.5$, há uma maior certeza que seja um bloco realmente não navegável. Para mostrar a melhoria tanto no mapa de navegabilidade quanto nas métricas de desempenho, a Figura 5.10 apresenta a nova classificação de dois *frames*. A análise completa do resultado da classificação do cenário I pode ser visualizado no apêndice D, e a Tabela 5.3 contém a análise do desempenho e melhoria do cenário I reavaliado.

Ao realizar o cálculo de desempenho sobre os *frames* reavaliados foi possível notar um ganho de desempenho em todas as métricas: acurácia, sensibilidade e precisão. A Figura 5.11, apresenta um gráfico com o aumento de desempenho obtido com a união entre os classificadores. O aumento foi de 3.85 % para a acurácia, 9.08 % para a sensibilidade e 1.2 % na precisão. Isto mostra como é complexa uma classificação de ambientes reais e como as influências em ambientes aquáticos estão presentes e devem ser tratadas com atenção.

Tabela 5.3 – Tabela de valores e métricas de desempenho com o segundo classificador

Frames	Verdadeiro Positivo	Verdadeiro Negativo	Falso Positivo	Falso Negativo	Acurácia	Sensibilidade	Precisão
Frame 1	371	374	10	13	97.01 %	96.61 %	97.38 %
Frame 2	383	374	10	1	98.57 %	99.74 %	97.46 %
Frame 3	375	383	1	9	98.70 %	97.66 %	99.73 %
Frame 4	372	379	5	12	97.79 %	96.88 %	98.67 %
Frame 5	369	383	1	15	97.92 %	96.09 %	99.73 %
Frame 6	351	415	1	1	99.74 %	99.72 %	99.72 %
Frame 7	351	382	2	33	95.44 %	91.41 %	99.43 %
Frame 8	352	399	17	0	97.79 %	100.00 %	95.39 %
Frame 9	294	435	13	26	94.92 %	91.88 %	95.77 %
Frame 10	220	432	80	36	84.90 %	85.94 %	73.33 %
Frame 11	216	500	44	8	93.23 %	96.43 %	83.08 %
Frame 12	306	414	48	0	93.75 %	100.00 %	86.44 %
Frame 13	335	409	23	1	96.88 %	99.70 %	93.58 %
Frame 14	355	371	29	13	94.53 %	96.47 %	92.45 %
Frame 15	347	395	15	11	96.61 %	96.93 %	95.86 %
Média do classificador:					95.85 %	96.36 %	93.86 %

5.3.3 Cenário II

A fim de avaliar o desempenho do método apresentado, foram feitos testes usando imagens de um segundo cenário. O cenário II também é um ambiente rural, mas apresenta menos vegetação e algumas casas que são um tipo de obstáculo para o veículo aquático. É possível analisar que no segundo cenário houve pouca incidência de reflexo do céu sobre a superfície, mas há um reflexo sutil da vegetação e das construções, como mostram as Figuras do apêndice D.

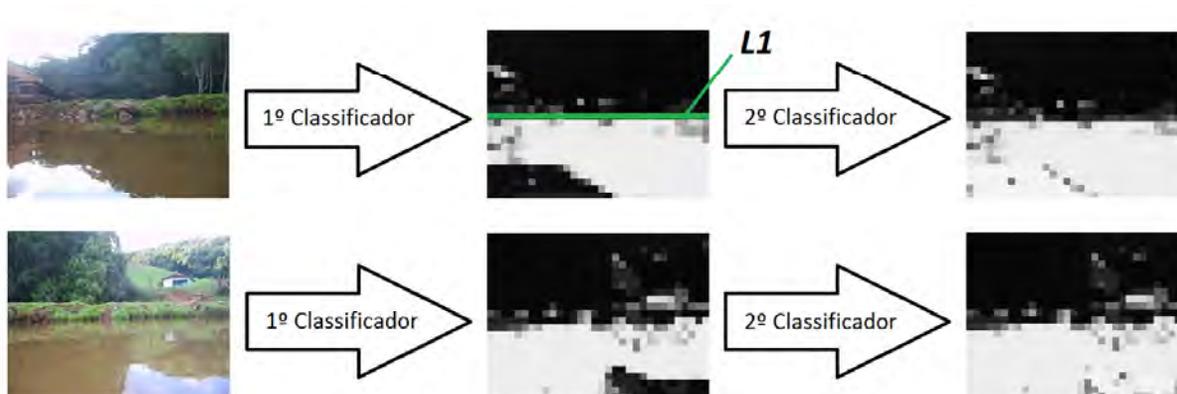


Figura 5.10 – Dois *frames* que mostram a melhoria no mapa de navegabilidade obtido pela união dos dois classificadores desenvolvidos.

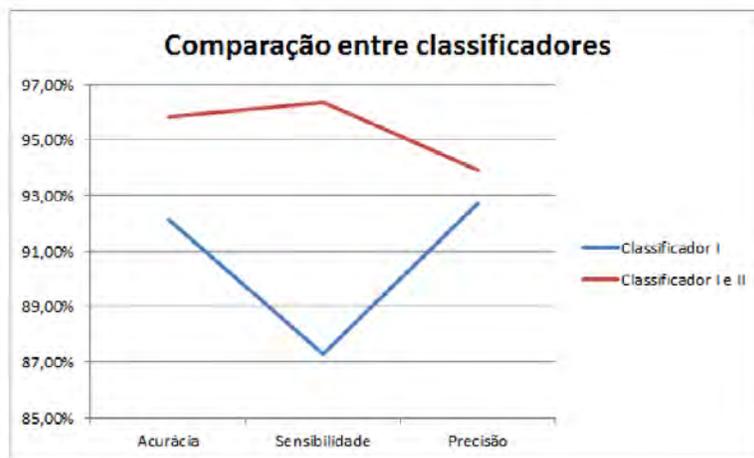


Figura 5.11 – Ganho de desempenho nas métricas avaliadas com a utilização dos dois classificadores no cenário I.

Usando os mesmos classificadores treinados para o cenário I e buscando mensurar o grau de adaptação do método apresentado, o veículo aquático foi inserido em um segundo cenário para coleta de uma sequência de vídeos. O objetivo era testar um novo cenário com maior diversidade e complexidade semelhante a uma possível aplicação real. Para este experimento se procurou aumentar a complexidade a partir de uma sequência de imagens em diferentes condições climáticas e de iluminação, através da coleta de imagens em diferentes datas e horários do dia, totalizando quatro vídeos. De cada vídeo foram extraídos seis *frames* aleatórios para classificação, totalizando um conjunto de 24 *frames*.

O apêndice D, apresenta os *frames* extraídos e o resultado obtido para os quatro vídeos do segundo cenário. A tabela 5.4, apresenta a métrica de desempenho para o cenário II, em que a média de acurácia, sensibilidade e precisão foi de 93.35 %, 96.26 %, 90.53 %, respectivamente.

5.3.4 Diversificando a entrada

Buscando aumentar ainda mais a complexidade dos cenários e avaliar a qualidade dos classificadores desenvolvidos, foram propostas algumas variações no conjunto de *frames* de entrada. Neste caso, foram usados cenários mais heterogêneos, com situações reais de possíveis aplicações de desastres naturais, tanto na etapa de classificação como no treinamento. Estas imagens correspondem ao cenário III.

O Cenário III corresponde a um ambiente urbano e descreve uma situação real de desastre natural. As imagens que compõem este cenário foram adquiridas durante um período de enchentes quando várias pessoas ficaram desabrigadas. Ambientes deste gênero

Tabela 5.4 – Tabela de valores e métricas de desempenho do classificador para o segundo cenário avaliado.

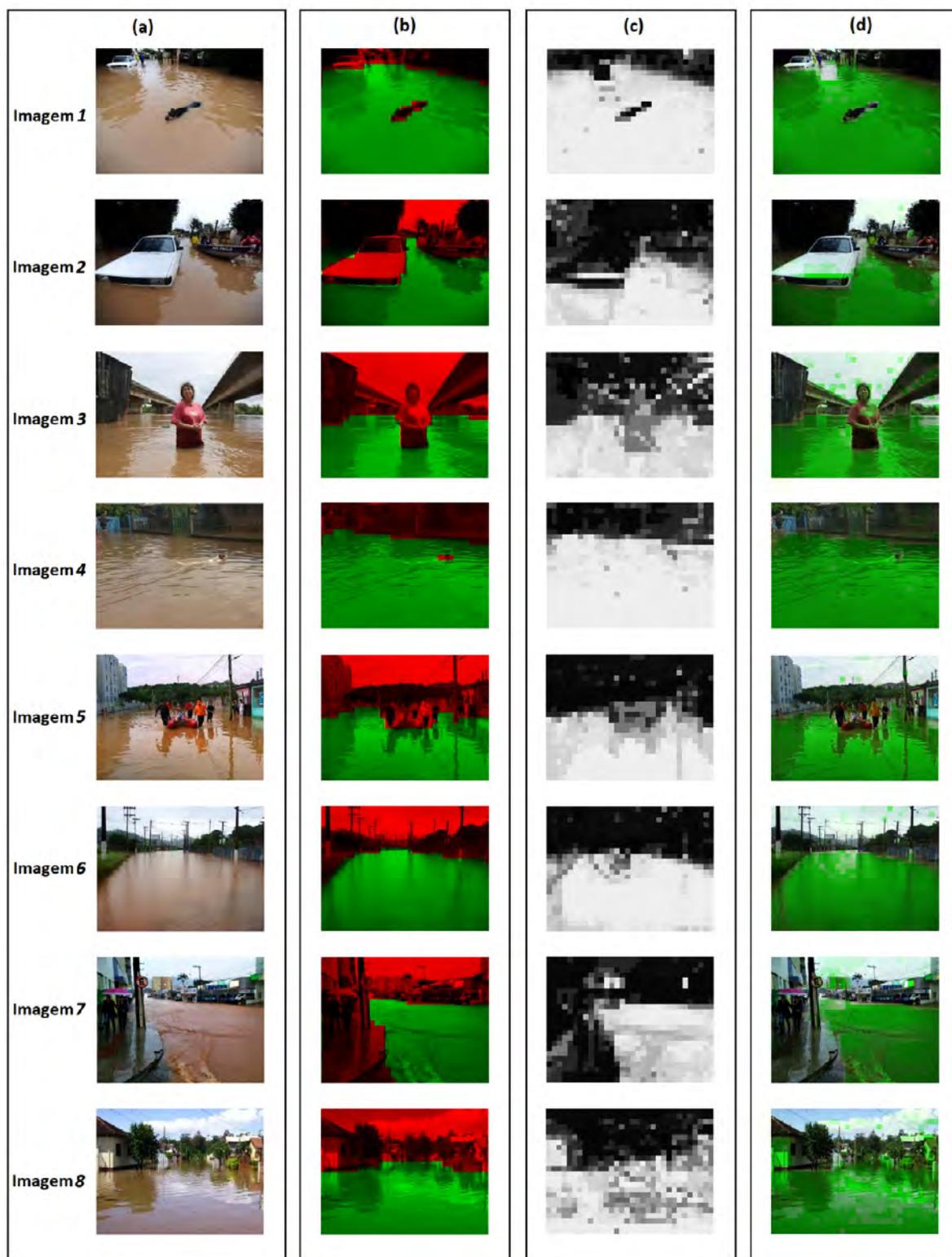
Frames	Verdadeiro Positivo	Verdadeiro Negativo	Falso Positivo	Falso Negativo	Acurácia	Sensibilidade	Precisão
<i>Frame 1-1</i>	315	430	3	20	97.01 %	94.03 %	99.06 %
<i>Frame 1-2</i>	502	239	19	8	96.48 %	98.43 %	96.35 %
<i>Frame 1-3</i>	317	436	1	14	98.05 %	95.77 %	99.69 %
<i>Frame 1-4</i>	352	349	60	7	91.28 %	98.05 %	85.44 %
<i>Frame 1-5</i>	308	365	88	7	87.63 %	97.78 %	77.78 %
<i>Frame 1-6</i>	300	447	1	20	97.27 %	93.75 %	99.67 %
<i>Frame 2-1</i>	309	429	19	11	96.09 %	96.56 %	94.21 %
<i>Frame 2-2</i>	451	286	7	24	95.96 %	94.95 %	98.47 %
<i>Frame 2-3</i>	297	440	20	11	95.96 %	96.43 %	93.69 %
<i>Frame 2-4</i>	302	447	12	7	97.53 %	97.73 %	96.18 %
<i>Frame 2-5</i>	309	416	36	7	94.40 %	97.78 %	89.57 %
<i>Frame 2-6</i>	304	375	88	1	88.54 %	98.61 %	77.55 %
<i>Frame 3-1</i>	305	403	45	15	92.19 %	95.31 %	87.14 %
<i>Frame 3-2</i>	350	348	57	13	90.89 %	96.42 %	86.00 %
<i>Frame 3-3</i>	382	348	22	16	95.05 %	95.98 %	94.55 %
<i>Frame 3-4</i>	362	303	86	17	86.59 %	95.51 %	80.80 %
<i>Frame 3-5</i>	383	336	34	15	93.62 %	96.23 %	91.85 %
<i>Frame 3-6</i>	309	365	89	5	87.76 %	98.41 %	77.64 %
<i>Frame 4-1</i>	403	297	49	19	91.15 %	95.50 %	89.16 %
<i>Frame 4-2</i>	315	416	26	11	95.18 %	96.63 %	92.38 %
<i>Frame 4-3</i>	427	325	6	10	97.92 %	97.71 %	98.61 %
<i>Frame 4-4</i>	398	337	24	9	95.70 %	97.79 %	94.31 %
<i>Frame 4-5</i>	302	440	8	18	96.61 %	94.38 %	97.42 %
<i>Frame 4-6</i>	325	302	107	34	81.64 %	90.53 %	75.23 %
Média do classificador:					93.35 %	96.26 %	90.53 %

são mais complexos, uma vez que apresentam uma grande diversidade pois podem conter pessoas, carros, animais e construções.

Para este experimento foram selecionadas 15 imagens de entrada que estão apresentadas na coluna (a) da Figura 5.12. Estas imagens são de diferentes situações e cenários, de forma a generalizar ao máximo o conjunto de entradas para o experimento. A coluna (b) apresenta a segmentação manual realizada sobre as imagens de entrada para a métrica de desempenho. Já a coluna (c), apresenta os resultados obtidos para cada imagem de entrada, mostrando de forma qualitativa o desempenho do classificador, em que é gerado o mapa de navegabilidade. Na coluna (d) é realizada uma sobreposição do mapa gerado sobre as imagens de entrada, de modo a sinalizar os blocos classificados como navegáveis.

A Tabela 5.5 apresenta a análise quantitativa sobre o conjunto de imagens de entrada, baseada nas segmentações realizadas na coluna (b) da Figura 5.12. O objetivo da

análise é calcular as métricas do desempenho do classificador sobre as diferentes situações do cenário III.



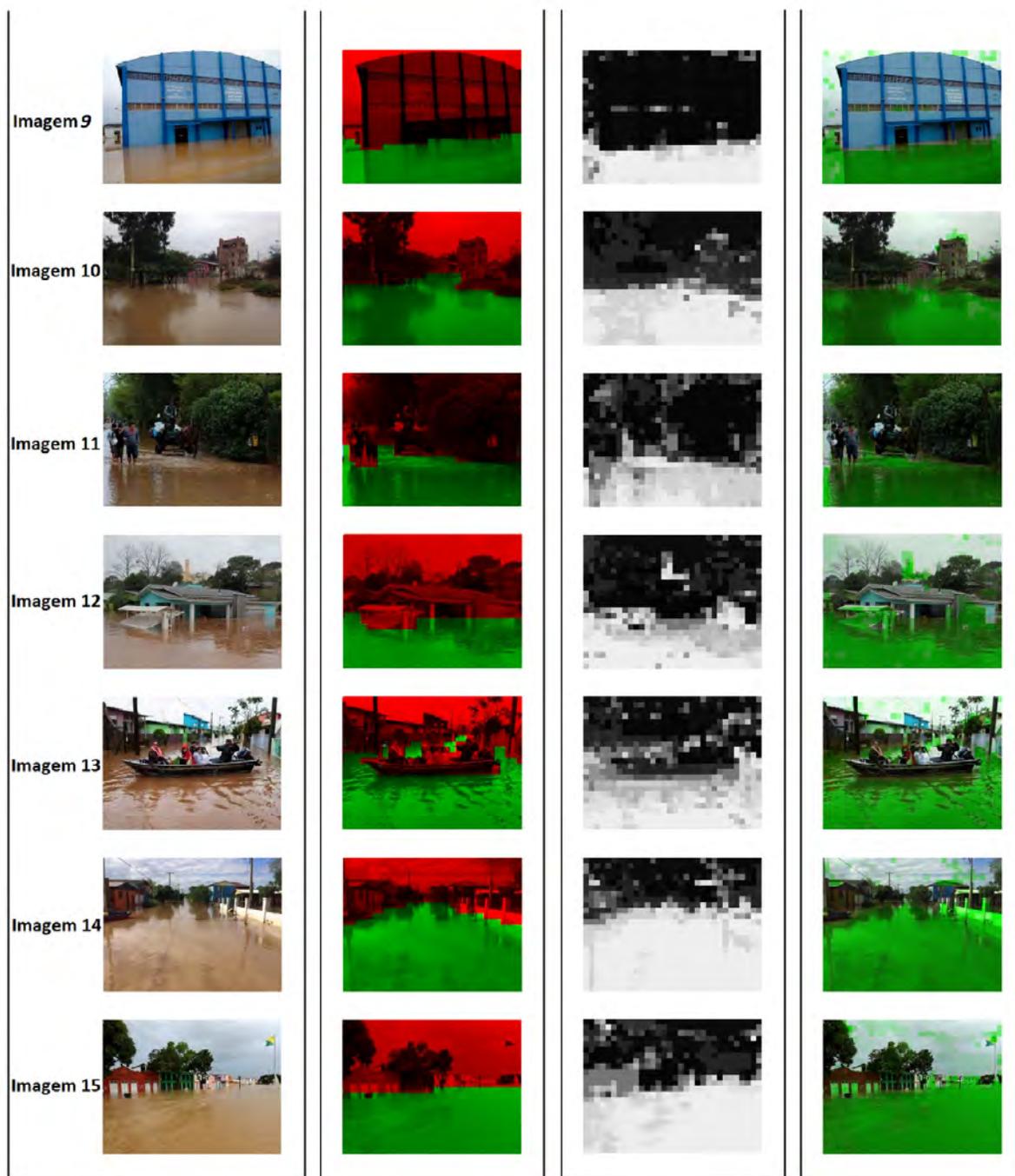


Figura 5.12 – Conjunto de imagens do cenário III que corresponde a uma situação real de desastre natural. A coluna (a) apresenta as imagens de entrada; (b) Segmentação realizada para métrica de desempenho; (c) O resultado obtido pelo classificador e (d) sobreposição da classificação sobre o conjunto de entrada [38].

Analisando as métricas de desempenho da Tabela 5.5 é possível notar que o classificador tem maiores índices de erro em valores de falso positivos. Entretanto, a métrica de precisão que é baseado nos valores de falso positivos obteve um desempenho acei-

Tabela 5.5 – Tabela de valores e métricas de desempenho do classificador, para o teste de variação das imagens de entrada apresentadas pelo cenário III.

Imagens	Verdadeiro Positivo	Verdadeiro Negativo	Falso Positivo	Falso Negativo	Acurácia	Sensibilidade	Precisão
Imagem 1	608	129	21	10	95.96 %	98.38 %	96.66 %
Imagem 2	327	362	39	40	89.71 %	89.10 %	89.34 %
Imagem 3	334	313	117	4	84.24 %	98.82 %	74.06 %
Imagem 4	546	193	21	8	96.22 %	98.56 %	96.30 %
Imagem 5	370	338	50	10	92.19 %	97.37 %	88.10 %
Imagem 6	451	288	21	8	96.22 %	98.26 %	95.55 %
Imagem 7	301	415	39	13	93.23 %	95.86 %	88.53 %
Imagem 8	409	262	92	5	87.37 %	98.79 %	81.64 %
Imagem 9	227	499	26	16	94.53 %	93.42 %	89.72 %
Imagem 10	332	390	26	20	94.01 %	94.32 %	92.74 %
Imagem 11	265	427	64	12	90.10 %	95.67 %	80.55 %
Imagem 12	270	418	78	2	89.58 %	99.26 %	77.59 %
Imagem 13	368	326	70	4	90.36 %	98.92 %	84.02 %
Imagem 14	446	234	83	5	88.54 %	98.89 %	84.31 %
Imagem 15	367	293	107	1	85.94 %	99.73 %	77.43 %
Média do classificador:					91.21 %	97.02 %	86.44 %

tável, acima de 86% na identificação das superfícies em média dos *frames* de entrada. Considera-se um bom desempenho, uma vez que é possível perceber através das imagens selecionadas a diversidade que um ambiente real pode apresentar, como vegetação, objetos, pessoas, construções entre outras.

5.3.5 Análise dos resultados e considerações

Ao analisar os resultados obtidos para cada cenário, é possível considerar que o método proposto alcançou uma taxa de precisão satisfatória nos testes e experimentos realizados. A Figura 5.13 apresenta uma síntese das métricas de desempenho avaliadas para todos os cenários classificados. Conseguiu-se uma acurácia média de 95,85%, com erro padrão de 0,924 para o cenário I, uma acurácia média de 93,35%, com erro padrão de 0,882 para o cenário II e uma acurácia média de 91,21%, com erro padrão de 0,980 para o cenário III. O gráfico também apresenta o erro padrão da média sobre cada métrica de desempenho e cenário, estimando a variabilidade entre as amostras.

Através das análises dos valores médios gerados para cada cenário, é possível verificar melhores resultados das métricas de desempenho avaliadas para o primeiro cenário. Uma razão para isso é devido ao fato de que algumas imagens adquiridas nesse cenário foram usadas para a formação e treinamento da RNA. O cenário II é bastante semelhante ao primeiro usado para treinamento. Assim, embora seja um cenário desconhecido, um

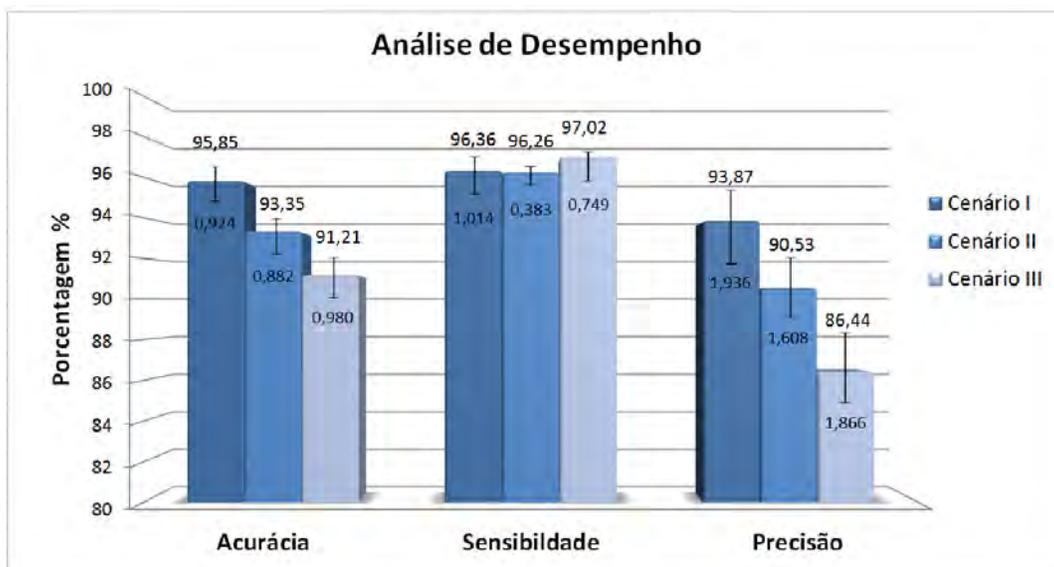


Figura 5.13 – Média dos resultados das avaliações para os três cenários e seus respectivos erros padrões.

bom resultado foi obtido com uma média superior a 90% entre as métricas avaliadas. O cenário III corresponde a um ambiente adverso, com muita diversidade, como pessoas, casas, carros e objetos flutuando na superfície aquática. Mesmo assim, o método provou ser eficiente, uma vez que tem uma boa avaliação de sensibilidade, o que demonstra um bom desempenho na identificação da superfície com uma alta taxa de valores verdadeiros positivos. Por outro lado, valores mais baixos para precisão aparecem devido à alta taxa de falsos positivos.

5.4 Embarcando o método

Com a finalidade de validar o método proposto e testar a sua aplicação em um ambiente real, todo software desenvolvido foi embarcado em um hardware. A aplicação do método desenvolvido em um hardware embarcado é uma etapa importante, uma vez que o poder de processamento em sistemas embarcados é limitado e muitas vezes é disputado com os demais sistemas tecnológicos implementados ao veículo. De modo a validar a etapa de aplicação do método apresentado, foi construído um protótipo e foram organizados e conduzidos testes para demonstrar possíveis aplicações reais do trabalho desenvolvido. A seguir é apresentado o hardware utilizado, bem como algumas especificações técnicas e testes de desempenho realizados sobre o protótipo.

5.4.1 Hardware

A escolha do hardware foi ponderada, uma vez que deve ser eficiente e possuir uma especificação de forma a suprir a necessidade do projeto, ou que cumpra as restrições necessárias. Desta forma, o hardware utilizado para teste foi a placa Raspberry Pi (RPi), modelo 2. Esta placa possui uma CPU quad-core ARM Cortex A7 de 900MHz, 1 GB de RAM e um processador gráfico 3D VideoCore IV. Inclui também quatro portas USB, quarenta pinos GPIO, porta HDMI e Ethernet. A Figura 5.14(a), mostra o componente descrito.

Além de possuir boas especificações, uma vantagem da placa RPi é os *shields* que complementam sua mobilidade e aplicação. Por exemplo, a Raspberry câmera, mostrada na figura 5.14(b) é um dispositivo eletrônico de pequenas dimensões, em torno de $25 \times 20 \times 9 \text{ mm}^3$, e que pesa 3g. A ligação com o hardware RPi é feita por meio de um cabo fita, que possui uma ligação direta ao processador da RPi através do barramento CSI, proporcionando uma maior largura de banda no transporte dos dados de pixel da câmara para o processador. Além disso, a câmera tem uma resolução máxima de 5 megapixels e uma lente de foco fixo a bordo.

A Figura 5.14(c) apresenta como foi projetada a construção do hardware e montagem da placa RPi e da câmera Pi ao veículo aquático para a realização dos testes e experimentos. A adaptação do hardware sobre o veículo se deu com o projeto e a impressão de peças em uma impressora 3D. Para a câmera Pi foi modelado um suporte para acoplagem e ajuste da altura, e para a RPi foi construída uma estrutura para fechamento e proteção do hardware.

O hardware executa a distribuição Raspbian Jessie, o sistema operacional da RPi baseado no Debian GNU/Linux. Inicialmente, para teste, foi usado um vídeo de entrada previamente coletado de um dos cenários avaliados, com o objetivo de analisar o comportamento e o desempenho do hardware. Os recursos monitorados foram a porcentagem do uso do processador, o uso de memória em execução e quantidade de FPS (*frames Per Second*) através da execução no hardware.

Ao executar os testes, algumas peculiaridades foram sendo analisadas. Por exemplo, apesar do hardware RPi possuir uma CPU quad-core, o sistema operacional da RPi modelo 2 não faz o gerenciamento do uso de cada núcleo. Desta forma, ao executar os primeiros testes os resultados foram de: 22% de uso do processador, 126.3 Megabyte de memória usada para a execução do método e 1.7 FPS. Sendo assim, para melhorar o desempenho e gerenciar o uso dos núcleos disponíveis no hardware, foi utilizada a biblioteca OpenMP de programação multi processo, com uma série de instruções executadas consecutivamente, que buscam melhorar o paralelismo a nível de *threads*.

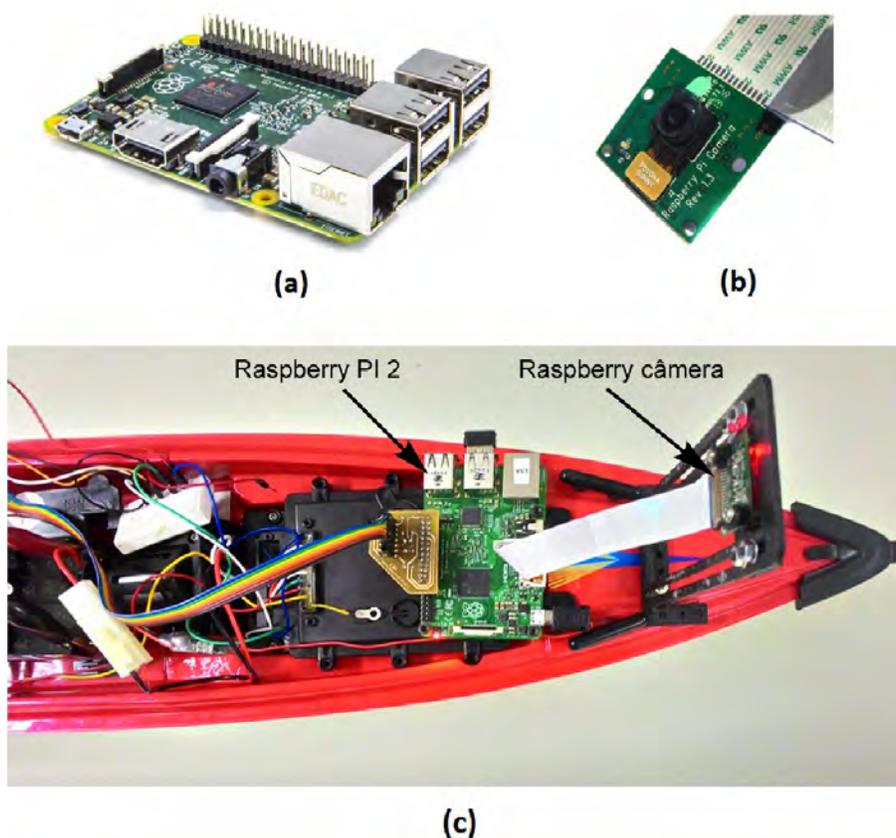


Figura 5.14 – Placa Raspberry Pi modelo 2 (a), *shield* da câmera Pi (b) e adaptação do *hardware* sobre o veículo aquático (c).

OpenMP

Com o objetivo de melhorar o processamento e gerenciamento dos núcleos sobre o hardware foi acrescentado o uso da biblioteca OpenMP. A biblioteca OpenMP permite passar um conjunto de diretivas ao compilador, rotinas de execução e variáveis de ambiente que otimizam o gerenciamento do uso das *threads*, que influenciam o comportamento no tempo de execução em programas escrito em C, C++ [7]. Depois de modelar a programação com o uso da biblioteca OpenMP, foram realizados novos testes para analisar o comportamento dos recursos do hardware. Assim, verificou-se que houve um ganho de desempenho sobre a quantidade de FPS calculados por segundos. A Figura 5.15, apresenta o ambiente Raspbian Jessie executando os testes com a modelagem final, demonstrando o melhor desempenho obtido sobre o hardware RPi. Os resultados alcançados foram de: 46% de uso do processador, 308.9 Megabyte de memória em execução e o FPS aumentou para 2.5. Para ilustrar estes resultados, os gráficos da Figura 5.16 apresentam a coleta dos resultados com uma comparação dos testes realizados entre o modo sequencial e paralelizado com o uso da biblioteca OpenMP.

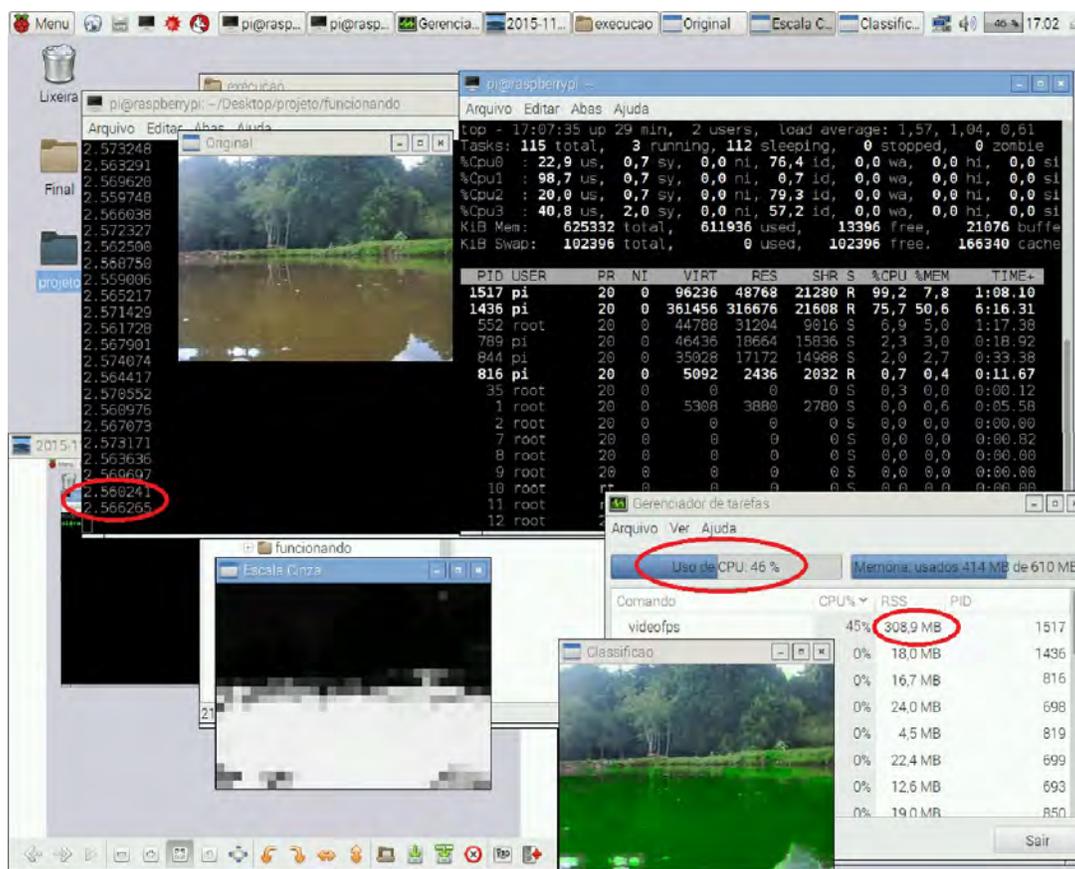


Figura 5.15 – Raspberry Pi 2, ambiente executando teste com a biblioteca OpenMP

Ao analisar os resultados obtidos sobre o desempenho, pode-se concluir que 2.5 FPS é considerado lento, mas aceitável, uma vez que o processamento está todo embarcado no veículo, sem a necessidade de uma comunicação ou processamento externo. Além disso, veículos aquáticos tratam-se de um tipo de locomoção que, em geral, apresenta velocidade baixa pela consequência da inércia oferecida pela água.

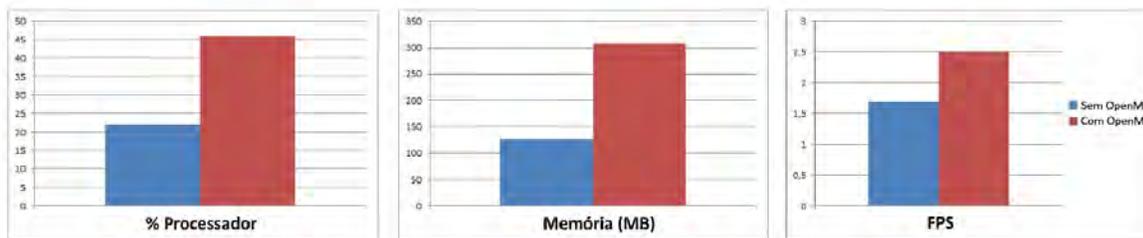


Figura 5.16 – Desempenho alcançado com o uso de OpenMP no hardware Raspberry Pi 2

5.4.2 Tomada de decisão na navegação

A avaliação e validação do algoritmo de navegação descrito na seção 4.5 foi feita através de simulação. Para esta avaliação, foram introduzidas 48 imagens com seus respectivos mapas de navegabilidade obtidos a partir dos cenários avaliados. Foram determinadas 12 imagens para cada possível comando de ação definido na máquina de estados finitos. De modo geral, os resultados foram satisfatórios, apesar de haverem situações complexas, com alguns mapas de navegabilidade com bastante ruído. A Tabela 5.6, apresenta o resultado obtido considerando os comandos esperados e os comandos executados pelo algoritmo. Os comandos esperados foram definidos manualmente de acordo com a melhor ação para a imagem em questão. Os comandos executados foram os resultados obtidos pelo algoritmo de navegação.

Tabela 5.6 – Tabela de valores esperados e obtidos na tomada de decisão na navegação.

Conjunto de comandos definidos	Comandos Esperados	Comando Executado
Ir para frente	12	8
Virar à Direita	12	9
Virar à esquerda	12	10
Parar	12	5
Média acerto de movimento	66,25%	

A Figura 5.17 apresenta alguns *frames* utilizados na realização dos testes e validação do algoritmo de navegação. Na Figura 5.17(a), são apresentados três *frames* que foram analisados de acordo com o esperado, isto é, o algoritmo realizou a melhor tomada de decisão para o estado atual do veículo. De forma oposta, nos *frames* da Figura 5.17(b) foram tomadas decisões incorretas, em que era esperado uma tomada de decisão e o algoritmo de navegação realizou outra ação.

O resultado do algoritmo de navegação com uma média de acerto de 66,25%, pode ser considerado satisfatório, principalmente devido ao fato que foram considerados apenas os valores esperados como corretos, mesmo que outros comandos também pudessem ser adequados em determinadas ações. O baixo acerto para o comando "parar" pode ter sido motivado pela definição da linha *L1*, pois o processo de decisão do veículo foi programado para "parar" apenas quando houvessem poucos ou nenhum bloco navegável à frente.

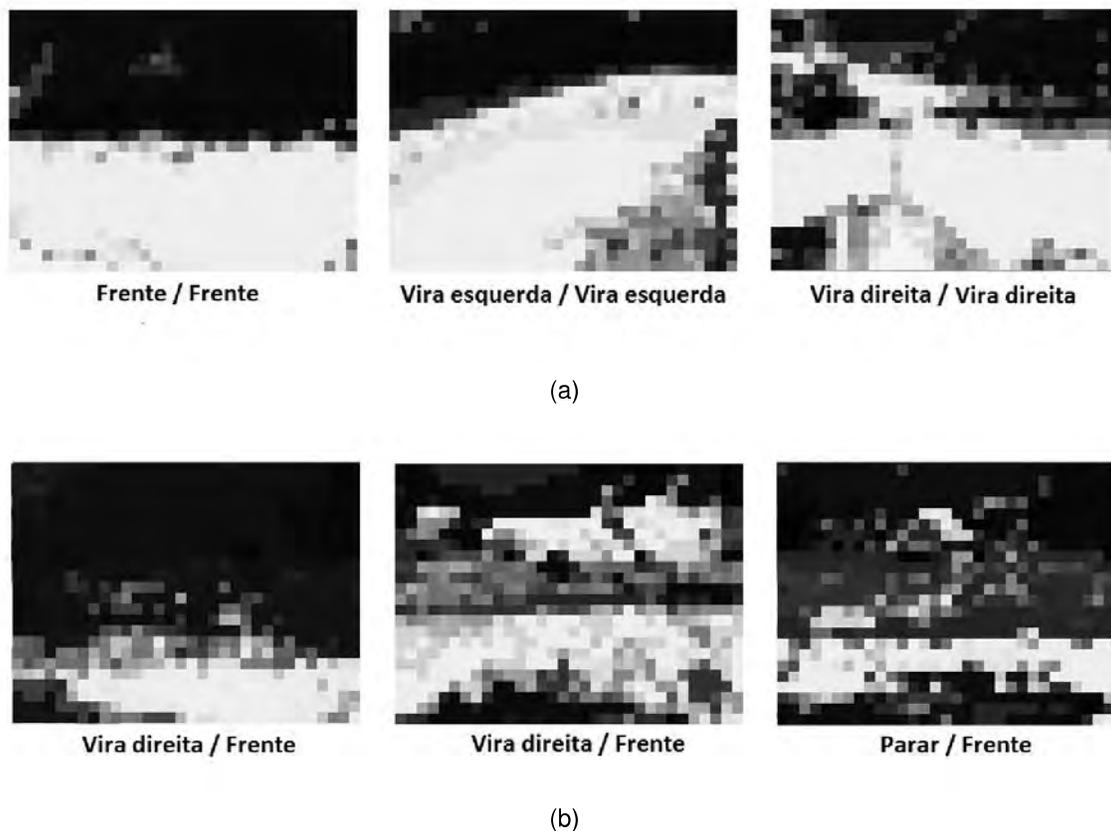


Figura 5.17 – Alguns *frames* nos quais foi tomada uma decisão correta pelo algoritmo de navegação (a), e incorreta (b), isto é, não de acordo com o esperado.

5.4.3 Protótipo

O protótipo foi construído com a finalidade de avaliar o método desenvolvido para navegação autônoma em ambiente real. Foi utilizada uma estrutura de um veículo aquático do tipo lancha de casco planante, que é sustentada pela água com movimentos de aquaplanagem. O hardware descrito na seção 5.4.1 e os demais dispositivos eletrônicos foram incorporados neste veículo aquático, como mostra a figura 5.18. Nesta figura é possível ver algumas imagens do veículo em uma situação real, através da qual foram realizados testes e avaliação de desempenho.

A construção do protótipo teve como finalidade demonstrar a aplicação do método proposto e avaliar o algoritmo de navegação que é baseado no mapa de navegabilidade gerado. Além disso, também havia a intenção de construir um veículo autônomo com processamento totalmente independente, ou seja, sem a necessidade de haver uma comunicação externa ou a execução de comandos por meio de um computador. Assim, por exemplo, este protótipo pode ser usado para navegar em ambientes desconhecidos ou pós-desastre.



(a)



(b)



(c)

Figura 5.18 – Protótipo desenvolvido em ação em um ambiente real.

6. CONCLUSÃO

Este trabalho teve como objetivo apresentar um método para identificação de superfícies aquáticas turvas, usando técnicas de visão computacional, verificando a viabilidade na construção e resolução do problema de mapeamento de ambientes não estruturados enfrentado por veículos aquáticos na navegação autônoma. Para isto foi desenvolvida uma metodologia baseada essencialmente em visão computacional com auxílio de métodos estatísticos multivariados de redução de dimensionalidade e redes neurais artificiais. Assim, foi construído um conjunto de classificadores que permitem transformar as informações visuais reais em um mapa de navegabilidade, através do qual é possível determinar com um grau de certeza se uma região é trafegável ou não. De posse do mapa de navegabilidade, foi aplicado um sistema de tomada de decisão para controle e tomada de ações. Todos estes métodos foram embarcados e testados em um protótipo de veículo aquático.

Esta solução foi assim desenvolvida para responder a questão de pesquisa colocada no capítulo 1: *Que tipo de informações podem ser coletadas das superfícies aquáticas turvas usando visão computacional, que aplicadas a um veículo autônomo podem auxiliá-lo na navegação neste ambiente?* Por isso, foram utilizadas propriedades óticas coletadas das superfícies aquáticas turvas para o desenvolvimento de uma técnica de visão computacional. Os resultados obtidos mostraram que este método é capaz de auxiliar a navegação de veículos aquáticos autônomos, dando autonomia suficiente para executar ações de navegação utilizando apenas informações visuais.

Para a coleta das imagens de entrada foi utilizada uma câmera monocular frontal. Várias abordagens foram estudadas a fim de conseguir aumentar o processamento de diferentes características de imagem, buscando a melhor forma de processar as imagens obtidas por esta câmera. A extração de características das imagens com a combinação de diferentes espaços de cor e atributos de textura eleva a capacidade de classificar as regiões navegáveis, mesmo em ambientes externos que sofrem mudanças de condições e variações inevitáveis. Foi possível observar que a união de um método estatístico para redução da dimensionalidade compactando os dados em componentes principais e posteriormente um método de aprendizado implementado com redes neurais artificiais, maximizaram o poder do classificador em diferenciar as regiões navegáveis das não navegáveis, mesmo em diferentes cenários.

A construção de dois classificadores com duas redes neurais, uma treinada e designada a classificar apenas possíveis reflexos sobre a superfície aquática, resultou em uma melhora no desempenho da classificação. Esta melhora se deu principalmente da região contendo reflexo, que também é a região de grande interesse para o veículo aquático, pois é comum ambientes aquáticos sofrerem com estas perturbações e dependências. Isso

complementou a modelagem final do classificador que teve maior robustez e qualidade no processo final de identificação.

O mapa de navegabilidade é o objetivo mais importante alcançado pelo trabalho, pois ele determina os limites de localização e distinção entre as regiões navegáveis e não navegáveis. Sua precisão demonstra a qualidade do método de identificação proposto, que apresentou bons resultados. No primeiro experimento de classificação realizado para o cenário I, as métricas de desempenho alcançadas foram de 95.85%, para acurácia, 96.36% na sensibilidade e 93.87% na precisão. Buscando demonstrar a capacidade de adaptação do classificador foram avaliados mais dois cenários. No cenário II, com várias tomadas e sequências de imagens de entrada foi obtido os valores de 95.35% de acurácia, 96.26% de sensibilidade e 90.53% de precisão. O cenário III, com imagens obtidas em situações de alagamento teve como resultado para acurácia, sensibilidade e precisão os valores de 91.21%, 97.02%, 86.44%, respectivamente.

Toda a metodologia desenvolvida e apresentada neste trabalho foi embarcada em um protótipo de veículo aquático, demonstrando ser um projeto viável para futura aplicação em situações de resgate e auxílio em desastres naturais. Este trabalho foi desenvolvido dentro do escopo de um projeto que tem como proposta o desenvolvimento de um sistema multi agente para prevenção e mitigação de desastres naturais. Portanto, espera-se que este trabalho possa ser incorporado neste projeto para auxiliar no deslocamento e navegação de veículos autônomos em ambientes pós-desastre.

6.1 Trabalhos futuros

Várias direções podem ser tomadas como trabalhos futuros, uma vez que o trabalho abordou vários métodos em conjunto para alcançar o objetivo proposto. Por exemplo, na seleção das características com o uso do PCA, por se tratar de um método estatístico exploratório multivariado, além da redução das variáveis ainda pode ser feito um estudo mais completo para calcular também o descarte de variáveis originais. Outra alternativa é explorar o cálculo de *loading* para melhorar os escores, uma vez que os escores fornecem a composição dos componentes em relação às amostras, enquanto o *loading* fornece essa mesma composição em relação às variáveis, com isto pode ser estimada influência de cada variável em cada amostra.

Considerando a pesquisa em visão computacional, é possível explorar outras técnicas de processamento de imagens. Como o método se baseia diretamente nas imagens de entrada, outros métodos ou técnicas de pré e pós processamento nestas imagens, podem gerar melhorias no classificador e conseqüentemente melhorar o resultado do mapa de navegabilidade. Por exemplo, de acordo com os experimentos realizados, a presença de reflexo interfere no desempenho do classificador, portanto um método de pré processa-

mento para diminuir os reflexos ajudariam na eficiência da classificação. Da mesma forma, um método de pós processamento no mapa de navegabilidade também ajudaria na qualidade do resultado dos classificadores, principalmente em relação a identificação da linha do horizonte. No mesmo sentido, o algoritmo de navegação poderia ser melhorado, principalmente na tomada de decisão baseado no estado "parar" uma vez que teve a menor taxa de acerto devido a sua dependência á linha do horizonte.

Além disso, outros sensores também poderiam ser incluídos no protótipo para auxiliar na tomada de decisão. Por exemplo, sensores de distância ou obstáculos ajudariam a aumentar o desempenho tanto na navegação quando na identificação da superfície navegável. A integração com outros sistemas aumenta a possibilidade e a capacidade de considerar informações adicionais, como altura de uma região, objetos presentes a frente do veículo em vez de apenas cores e texturas para classificar as regiões entre navegáveis e não navegáveis.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Achar, S.; Sankaran, B.; Nuske, S.; Scherer, S.; Singh, S. “Self-supervised segmentation of river scenes”. In: IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 6227–6232.
- [2] Barreto, J. M. “Introdução as redes neurais artificiais”, Relatório Técnico, Departamento de Informática e de Estatística - Universidade Federal de Santa Catarina - UFSC, 2002, 57p.
- [3] Bradski, G.; Kaehler, A. “Learning OpenCV: Computer vision with the OpenCV library”. O’Reilly Media, Inc., 2008, 580p.
- [4] Brumatti, M. “Redes neurais artificiais”, Relatório Técnico, Universidade Federal do Espírito Santo - UFES, 2005, 8p.
- [5] Centro Universitário de Estudo e Pesquisa sobre Desastres, C. U. “Atlas brasileiro de desastres naturais – 1991 a 2012”, *Revista Volume Brasil*, vol. 2 rev. ampl., 2013, pp. 126.
- [6] Chambers, A.; Achar, S.; Nuske, S.; Rehder, J.; Kitt, B.; Chamberlain, L.; Haines, J.; Scherer, S.; Singh, S. “Perception for a river mapping robot”. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011, pp. 227–234.
- [7] Chapman, B.; Jost, G.; van der Pas, R. “Using OpenMP: Portable Shared Memory Parallel Programming”. MIT Press, 2008, 353p.
- [8] de Almeida, A. R. “Redes neurais”, Relatório Técnico, Universidade Federal do Espírito Santo - UFES, Vitória, 2012, 31p.
- [9] Dias, N.; Almeida, C.; Ferreira, H.; Almeida, J.; Martins, A.; Dias, A.; Silva, E. “Manoeuvre based mission control system for autonomous surface vehicle”. In: OCEANS-EUROPE, 2009, pp. 1–5.
- [10] do Carmo, R. L.; Anazawa, T. M. “Mortalidade por desastres no brasil: o que mostram os dados”, *Revista Ciência & Saúde Coletiva*, vol. 19–9, 2014.
- [11] Dunbabin, M.; Grinham, A. “Experimental evaluation of an autonomous surface vehicle for water quality and greenhouse gas emission monitoring”. In: International Conference on Robotics and Automation (ICRA), 2010, pp. 5268–5274.
- [12] Fann. “Fast artificial neural network library”. Capturado em: <http://leenissen.dk/fann/wp/>, Maio 2015.

- [13] Fawcett, T. “An introduction to roc analysis”, *Pattern recognition letters*, vol. 27–8, 2006, pp. 861–874.
- [14] Fernandes, L. G. L.; Portugal, M. S.; Navaux, P. O. “Previsão de séries de tempo: redes neurais artificiais e modelos estruturais”, *Pesquisa e Planejamento Econômico*, vol. 26–2, 1996, pp. 253–276.
- [15] Forsyth, D.; Ponce, J. “Computer Vision: A Modern Approach”. Prentice Hall, 2003, 693p.
- [16] Gong, X.; Subramanian, A.; Wyatt, C. L. “A two-stage algorithm for shoreline detection”. In: *Applications of Computer Vision (WACV)*, 2007, pp. 40–40.
- [17] Gonzalez, R. C.; Woods, R. E. “Digital image processing”. Pearson Education, 2011, 976p.
- [18] Haykin, S. “Neural Networks: A Comprehensive Foundation”. Macmillan, 1994, 696p.
- [19] Haykin, S. S. “Redes neurais”. Bookman, 2001, 900p.
- [20] Heidarsson, H. K.; Sukhatme, G. “Obstacle detection from overhead imagery using self-supervised learning for autonomous surface vehicles”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 3160–3165.
- [21] Hornik, K.; Stinchcombe, M.; White, H. “Multilayer feedforward networks are universal approximators”, *Neural networks*, vol. 2–5, 1989, pp. 359–366.
- [22] Huntsberger, T.; Aghazarian, H.; Howard, A.; Trotz, D. C. “Stereo vision–based navigation for autonomous surface vessel”, *Journal of Field Robotics*, vol. 28–1, 2011, pp. 3–18.
- [23] Iqbal, M.; Morel, O.; Meriaudeau, F. “A survey on outdoor water hazard detection”, *International Conference on Information Communication Technology and Systems*, vol. 5–C07, 2009, pp. 33–40.
- [24] Jolliffe, I. “Principal component analysis”. Wiley Online Library, 2002, 271p.
- [25] Kron, W. “Flood disasters—a global perspective”, *Water Policy*, vol. 17–S1, 2015, pp. 6–24.
- [26] Laws, K. I. “Rapid texture identification”. In: *24th Annual Technical Symposium*, 1980, pp. 376–381.
- [27] Lima, V. R. d. “Desenvolvimento e avaliação de sistema neural para redução de tempo de ensaios de desempenho de compressores”, *Dissertação de Mestrado*, Universidade Federal de Santa Catarina – UFSC, Florianópolis, SC, Brasil, 2010, 173p.

- [28] Mallot, H. A. "Computational vision: information processing in perception and visual behaviour". MIT Press, 2000, 296p.
- [29] Masmitja, I.; Masmitja, G.; Gonzalez, J.; Shariat-Panahi, S.; Gomariz, S. "Development of a control system for an autonomous underwater vehicle". In: IEEE/OES Autonomous Underwater Vehicles, 2010, pp. 1–5.
- [30] Montgomery, E.; Ludwig, O. "Redes neurais: fundamentos e aplicações com programas em C". Ciência Moderna, 2007, 125p.
- [31] Morais, J. T. G. "Análise de componentes principais integrada a redes neurais artificiais para predição de matéria orgânica.", Dissertação de Mestrado, Universidade Federal da Bahia – UFBA, Salvador, BA, Brasil, 2011, 68p.
- [32] Murphy, R. R.; Steimle, E.; Griffin, C.; Cullins, C.; Hall, M.; Pratt, K. "Cooperative use of unmanned sea surface and micro aerial vehicles at hurricane wilma", *Journal of Field Robotics*, vol. 25–3, 2008, pp. 164–180.
- [33] Murtagh, F. "Principal components analysis in c". Capturado em: <http://ftp.uni-bayreuth.de/math/statlib/multi/pca.c>, Nov 2014.
- [34] Oliveira, S. C. "Desenvolvimento de uma rede neural artificial para detecção de descargas parciais em cadeias de isoladores de linhas de alta tensão", Relatório Técnico, Universidade de Pernambuco - UPE, 2009, 50p.
- [35] Perelmuter, G.; Carrera, E.; Vellasco, M.; Pacheco, A. "Reconhecimento de imagens bidimensionais utilizando redes neurais artificiais". In: Simpósio Brasileiro de Computação Gráfica e Processamento de Imagens (SIBGRAPI), 1995, pp. 197–203.
- [36] Rankin, A.; Matthies, L. "Daytime water detection based on color variation". In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2010, pp. 215–221.
- [37] Rankin, A. L.; Matthies, L. H.; Bellutta, P. "Daytime water detection based on sky reflections". In: IEEE International Conference on Robotics and Automation (ICRA), 2011, pp. 5329–5336.
- [38] RBS, A. "Notícias zero hora temporal na capital". Capturado em: <http://zh.clicrbs.com.br/rs/>, Maio 2015.
- [39] Romero, M.; Guo, Y.; Ieng, S.-H.; Plumet, F.; Benosman, R.; Gas, B. "Omni-directional camera and fuzzy logic path planner for autonomous sailboat navigation". In: Iberoamerican Conference on Electronics Engineering and Computer Science, 2011, pp. 50.

- [40] Santana, P.; Mendonça, R.; Barata, J. “Water detection with segmentation guided dynamic texture recognition”. In: IEEE International Conference on Robotics and Biomimetics (ROBIO), 2012, pp. 1836–1841.
- [41] SBC. “Veículos autônomos não tripulados: Conheça esta nova tecnologia que começa a ganhar força no Brasil.”, *Revista da Sociedade Brasileira de Computação*, vol. 1–24, 2014.
- [42] Scalco, R. “Introdução à Computação Gráfica”. Roberto Scalco, 2005, 175p.
- [43] Scerri, P.; Kannan, B.; Velagapudi, P.; Macarthur, K.; Stone, P.; Taylor, M.; Dolan, J.; Farinelli, A.; Chapman, A.; Dias, B.; et al.. “Flood Disaster Mitigation: A Real-World Challenge Problem for Multi-agent Unmanned Surface Vehicles”. Springer Berlin Heidelberg, 2012, cap. Flood Disaster Mitigation: A Real-World Challenge Problem for Multi-agent Unmanned Surface Vehicles, pp. 252–269.
- [44] Shannon, C. E. “A mathematical theory of communication”, *Bell Syst. Tech. J.*, vol. 27, 1948, pp. 379–423, 623–656.
- [45] Shinzato, P.; Grassi, V.; Osorio, F.; Wolf, D. “Fast visual road recognition and horizon detection using multiple artificial neural networks”. In: IEEE Intelligent Vehicles Symposium, 2012, pp. 1090–1095.
- [46] Shinzato, P. Y. “Sistema de identificação de superfícies navegáveis baseado em visão computacional e redes neurais artificiais”, Dissertação de Mestrado, Instituto de Ciências Matemáticas e de Computação – ICMC–USP, São Carlos, SP, Brasil, 2010, 96p.
- [47] Shinzato, P. Y.; Wolf, D. F. “A road following approach using artificial neural networks combinations”, *Journal of Intelligent & Robotic Systems*, vol. 62–3-4, 2011, pp. 527–546.
- [48] Silva, A. P. D. “Efficient variable screening for multivariate analysis”, *Journal of Multivariate Analysis*, vol. 76–1, 2001, pp. 35–62.
- [49] Silva, E.; Oliveira, A. C. d. “Dicas para a configuração de redes neurais”, Relatório Técnico, Universidade Federal do Rio de Janeiro - UFRJ, 2001, 4p.
- [50] Silva, F. C. d.; Costa, R. M. “Um sistema computacional para extração de características faciais”, Relatório Técnico, Fundação Universidade Federal do Rio Grande - FURG, 2007, 82p.
- [51] Snyder, F. D.; Morris, D. D.; Haley, P. H.; Collins, R. T.; Okerholm, A. M. “Autonomous river navigation”. In: SPIE, Mobile Robots, 2004, pp. 221–232.

- [52] Staff, J. M. S.; Hall, P. "Applied Multivariate Statistical Analysis". Prentice Hall PTR, 1992, 208p.
- [53] Steimle, E. T.; Murphy, R. R.; Lindemuth, M.; Hall, M. L. "Unmanned marine vehicle use at hurricanes wilma and ike". In: OCEANS, 2009, pp. 1–6.
- [54] Tchemra, A. H. "Aplicação da tecnologia adaptativa em sistemas de tomada de decisão", *Revista IEEE América Latina*, vol. 5–7, 2007, pp. 1548–0992.
- [55] (UNISDR). "The united nations office for disaster risk reductio". Capturado em: <http://www.unisdr.org/we/inform/disaster-statistics>, Novembro 2014.
- [56] Wang, J.; Gu, W.; Zhu, J. "Design of an autonomous surface vehicle used for marine environment monitoring". In: International Conference on Advanced Computer Control, 2009, pp. 405–409.
- [57] Wang, M.; Lai, C. "A Concise Introduction to Image Processing using C++". CRC Press, 2016, 264p.
- [58] Wold, S.; Esbensen, K.; Geladi, P. "Principal component analysis", *Chemometrics and intelligent laboratory systems*, vol. 2–1, 1987, pp. 37–52.
- [59] Yang, J.; Rao, D.; Chung, S.; Hutchinson, S. "Monocular vision based navigation in gps-denied riverine environments". In: American Institute of Aeronautics and Astronautics (AIAA), 2011, pp. 1–12.
- [60] Yao, T.; Xiang, Z.; Liu, J.; Xu, D. "Multi-feature fusion based outdoor water hazards detection". In: International Conference on Mechatronics and Automation (ICMA), 2007, pp. 652–656.
- [61] Yu, L.; Wang, S.; Lai, K. K. "An integrated data preparation scheme for neural network data analysis", *IEEE Transactions on Knowledge and Data Engineering*, vol. 18–2, 2006, pp. 217–230.
- [62] Zhang, J.; Morris, A. J. "A sequential learning approach for single hidden layer neural networks", *Neural networks*, vol. 11–1, 1998, pp. 65–80.

APÊNDICE A – TOTAL DE CARACTERÍSTICAS EXTRAÍDAS PARA GERAÇÃO DOS ATRIBUTOS

Tabela A.1: Tabela contendo os atributos calculados

Atributo	Descrição do atributo
1	Média do canal de cor R no espaço RGB
2	Média do canal de cor G no espaço RGB
3	Média do canal de cor B no espaço RGB
4	Média do canal de cor H no espaço HSV
5	Média do canal de cor S no espaço HSV
6	Média do canal de cor V no espaço HSV
7	Média do canal de cor Y no espaço YUV
8	Média do canal de cor U no espaço YUV
9	Entropia do canal de cor R no espaço RGB
10	Entropia do canal de cor G no espaço RGB
11	Entropia do canal de cor B no espaço RGB
12	Entropia do canal de cor H no espaço HSV
13	Entropia do canal de cor S no espaço HSV
14	Entropia do canal de cor V no espaço HSV
15	Entropia do canal de cor Y no espaço YUV
16	Entropia do canal de cor U no espaço YUV
17	Variância do canal de cor R no espaço RGB
18	Variância do canal de cor G no espaço RGB
19	Variância do canal de cor B no espaço RGB
20	Variância do canal de cor H no espaço HSV
21	Variância do canal de cor S no espaço HSV
22	Variância do canal de cor V no espaço HSV
23	Variância do canal de cor Y no espaço YUV
24	Variância do canal de cor U no espaço YUV
25	Energia do canal de cor R no espaço RGB
26	Energia do canal de cor G no espaço RGB
27	Energia do canal de cor B no espaço RGB
28	Energia do canal de cor H no espaço HSV
29	Energia do canal de cor S no espaço HSV
30	Energia do canal de cor V no espaço HSV
31	Energia do canal de cor Y no espaço YUV

Continua na próxima página

Tabela A.1 – Continuação da página anterior

Atributo	Descrição do atributo
32	Energia do canal de cor U no espaço YUV

APÊNDICE B – COLEÇÃO DE *FRAMES* DE TREINAMENTO

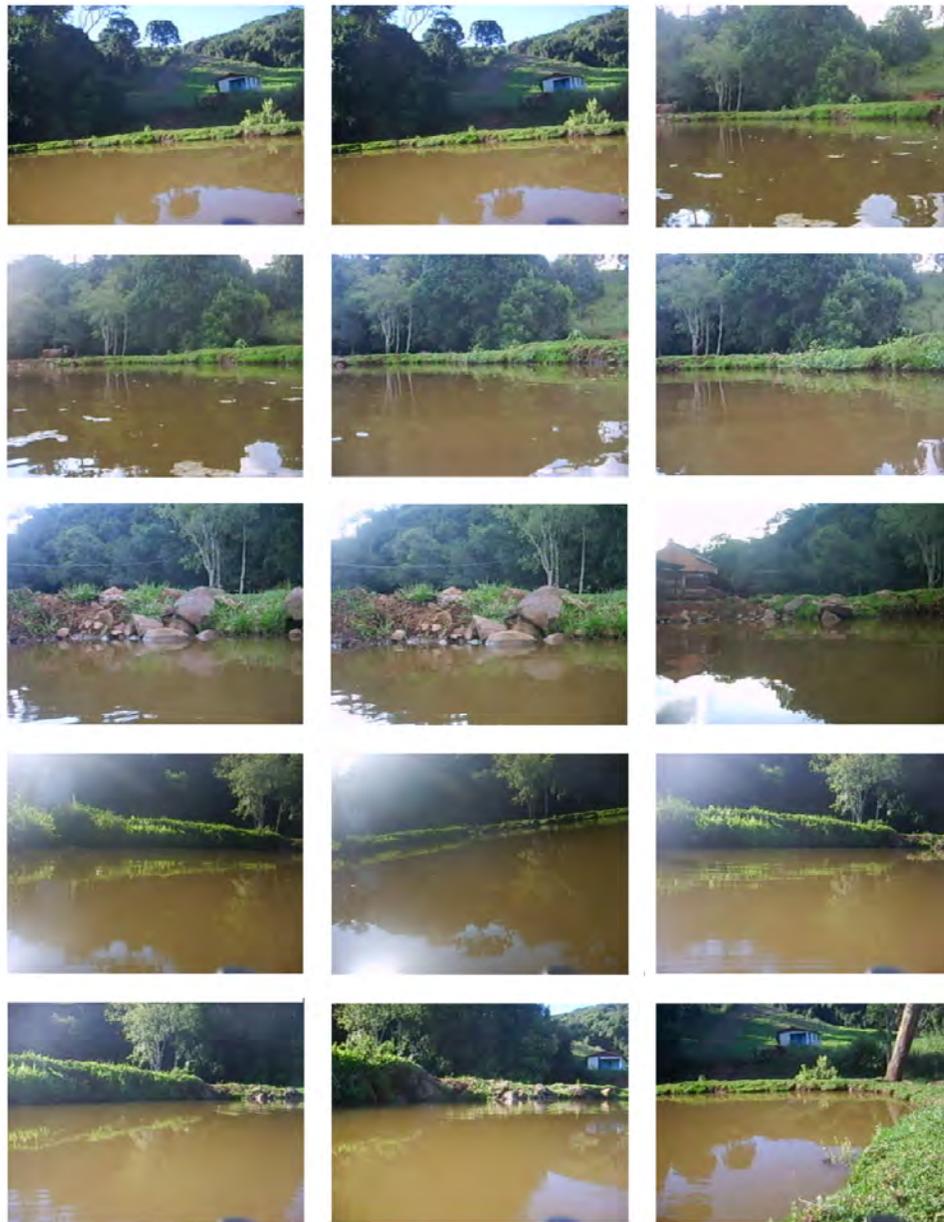
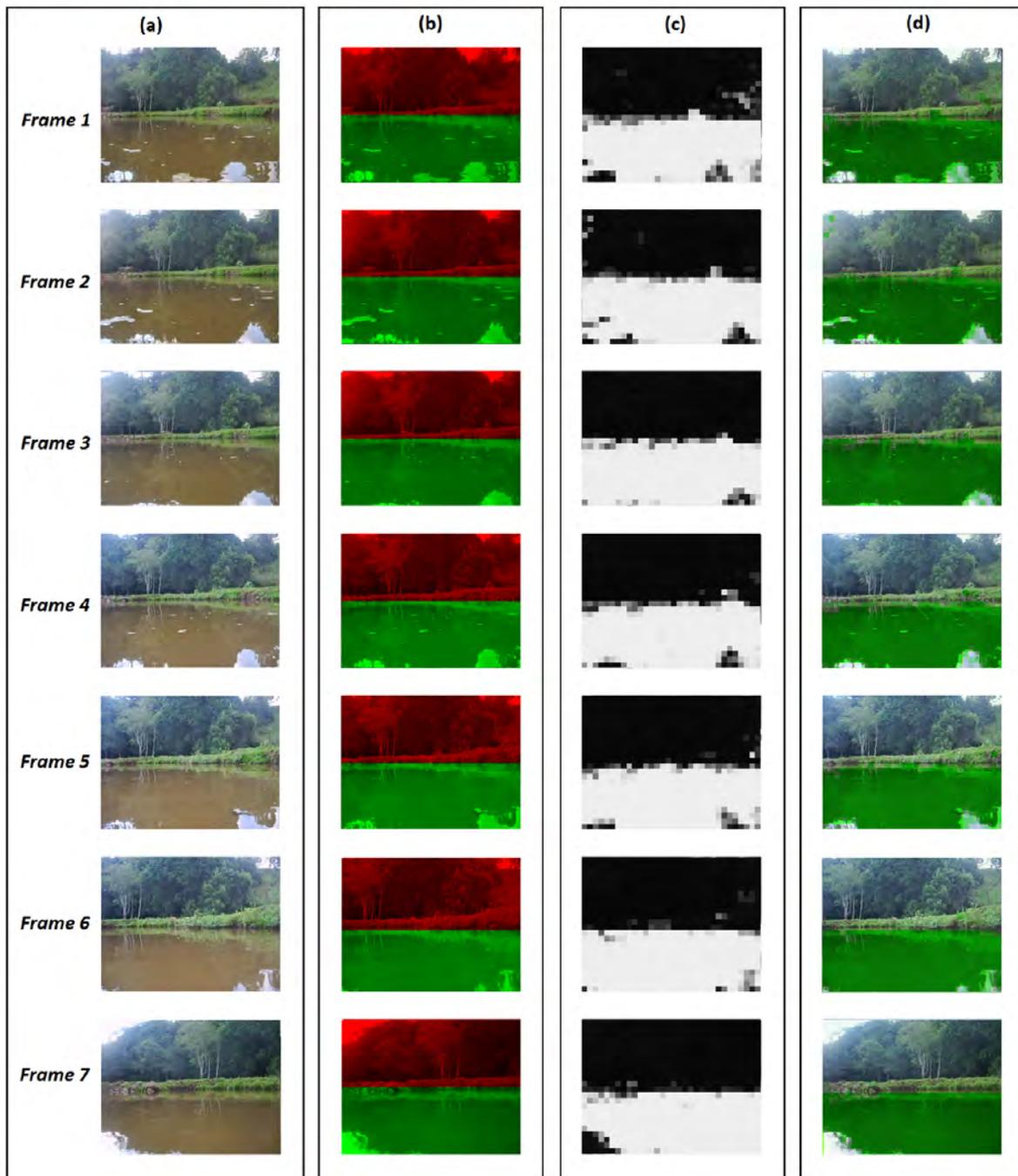


Figura B.1 – Conjunto de *frames* utilizados para treinamento.

APÊNDICE C – RESULTADOS OBTIDOS COM A PRIMEIRA CLASSIFICAÇÃO



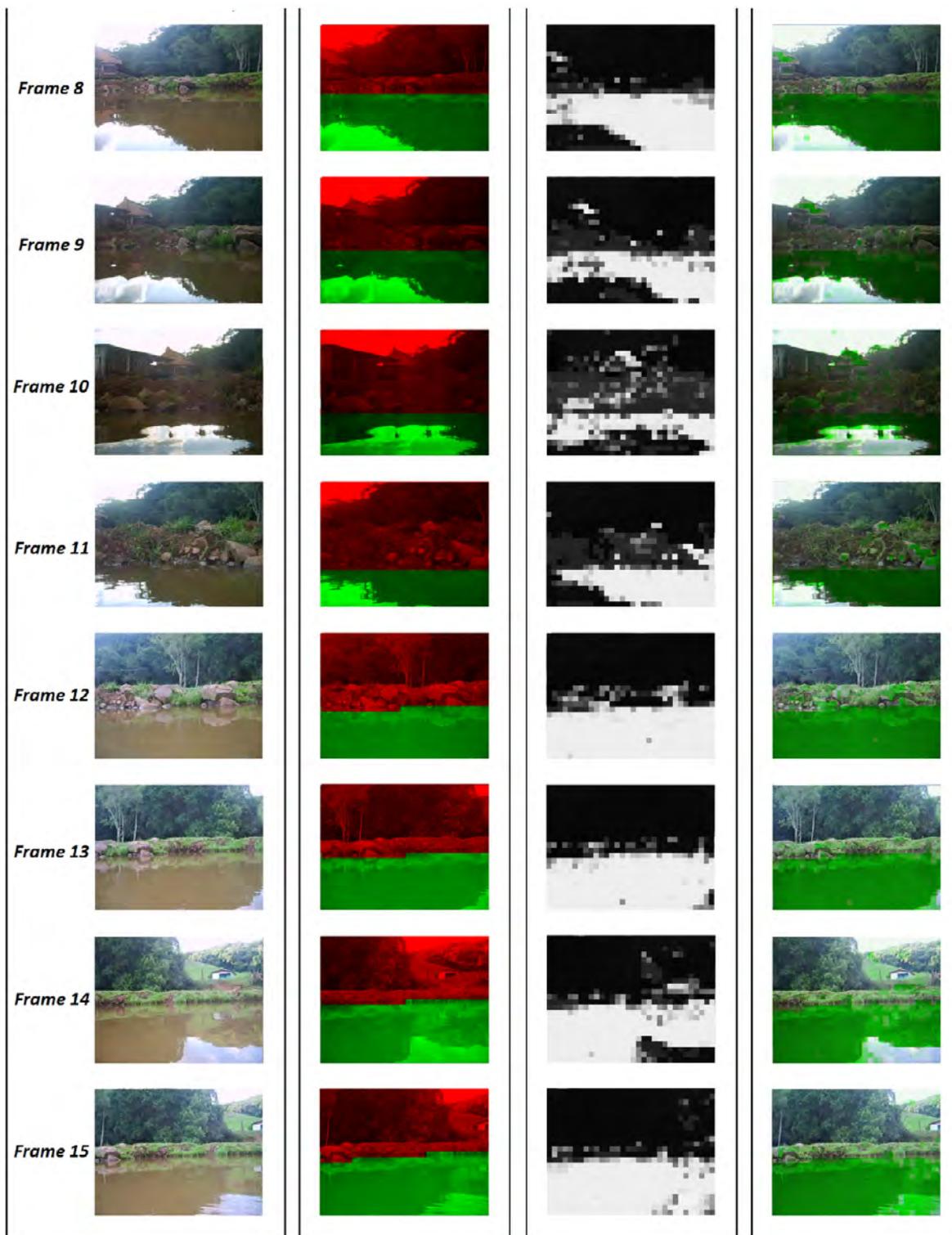
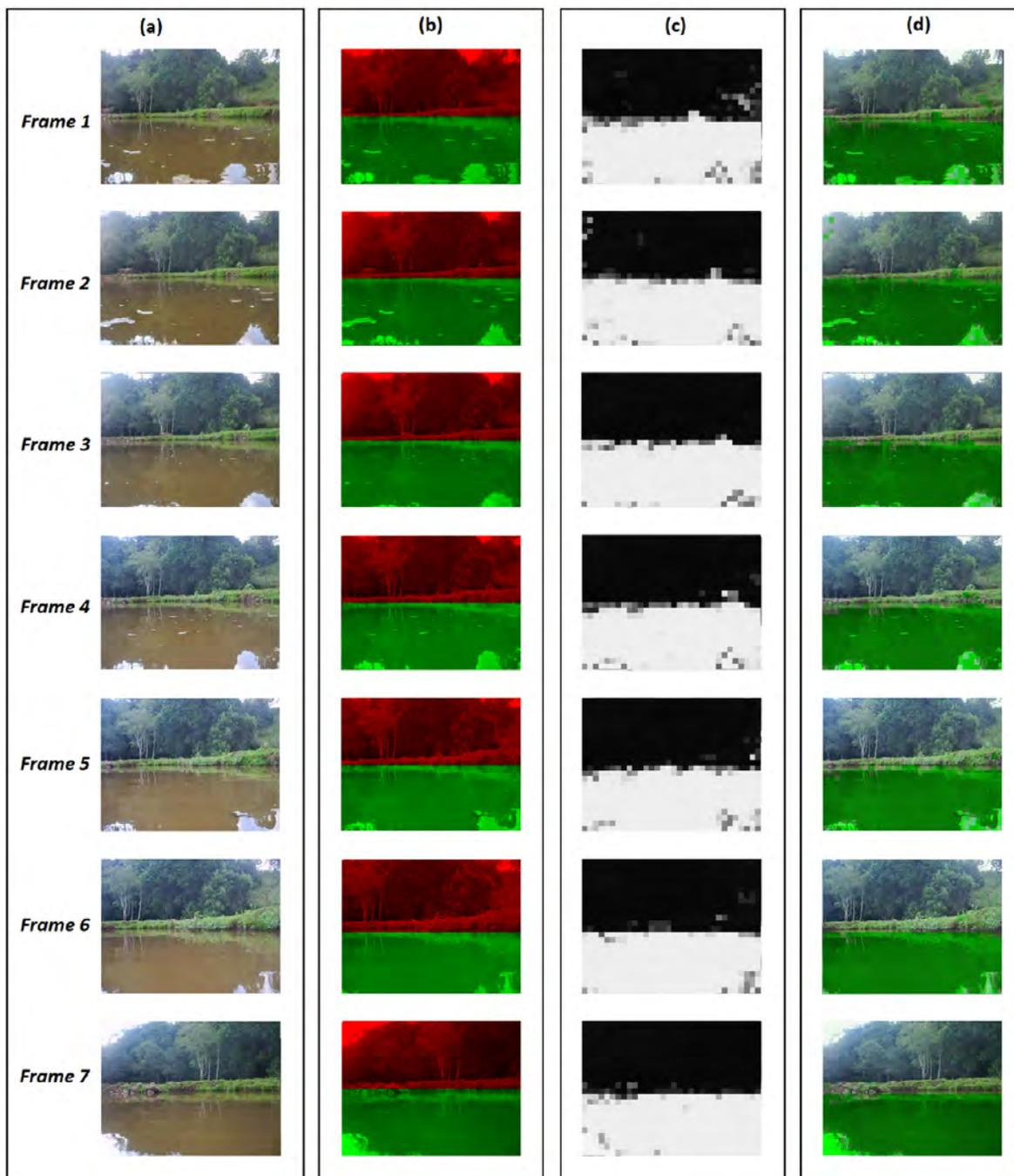


Figura C.1 – Conjunto de *frames* extraídos para classificação e experimentos. Em (a) apresenta os *frames* de entrada; (b) Segmentação manual realizada; (c) O resultado obtido; E (d) União da classificação com a imagem de entrada.

**APÊNDICE D – RESULTADOS OBTIDOS COM A NOVA
CLASSIFICAÇÃO DO CENÁRIO I, UTILIZANDO O SEGUNDO
CLASSIFICADOR**



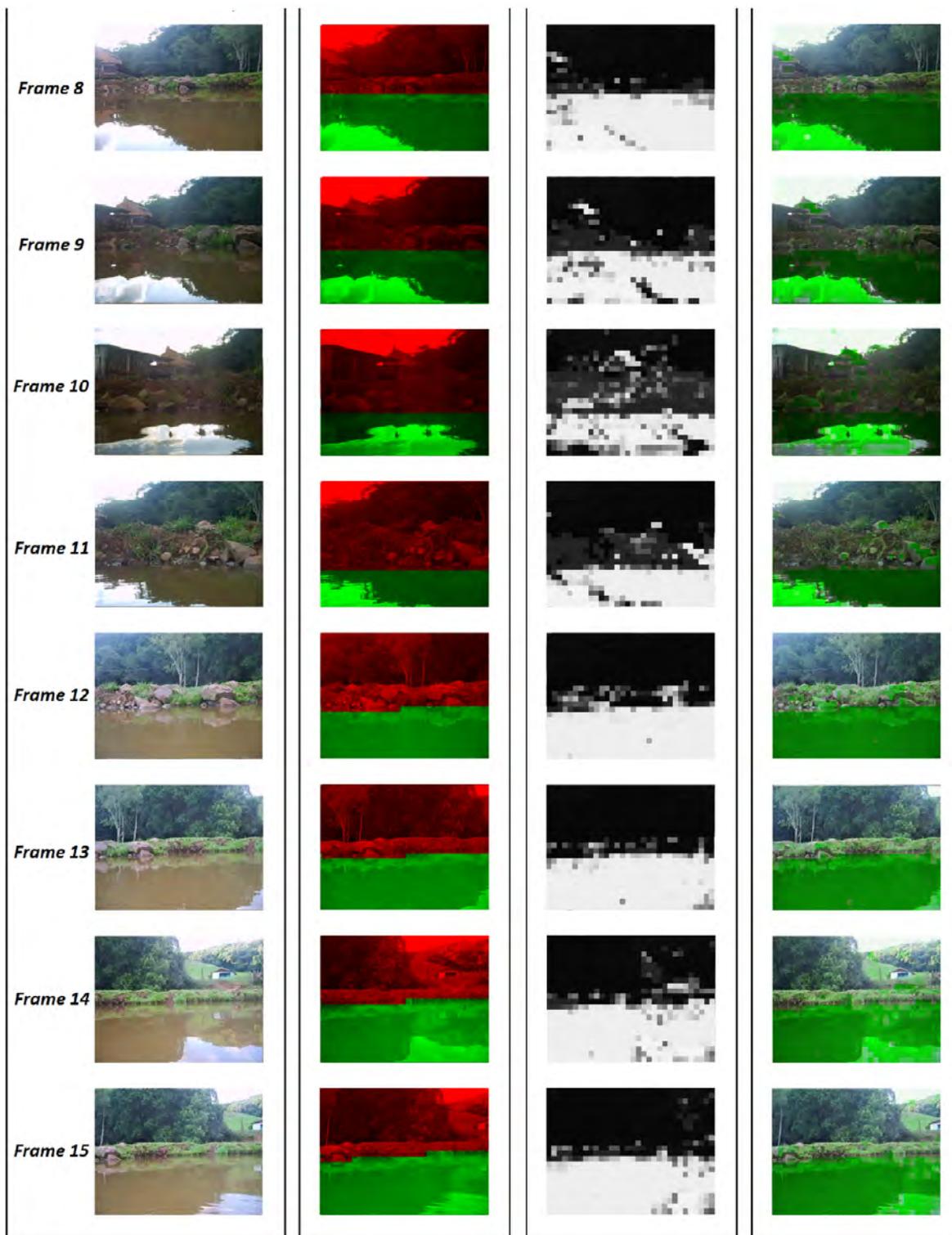
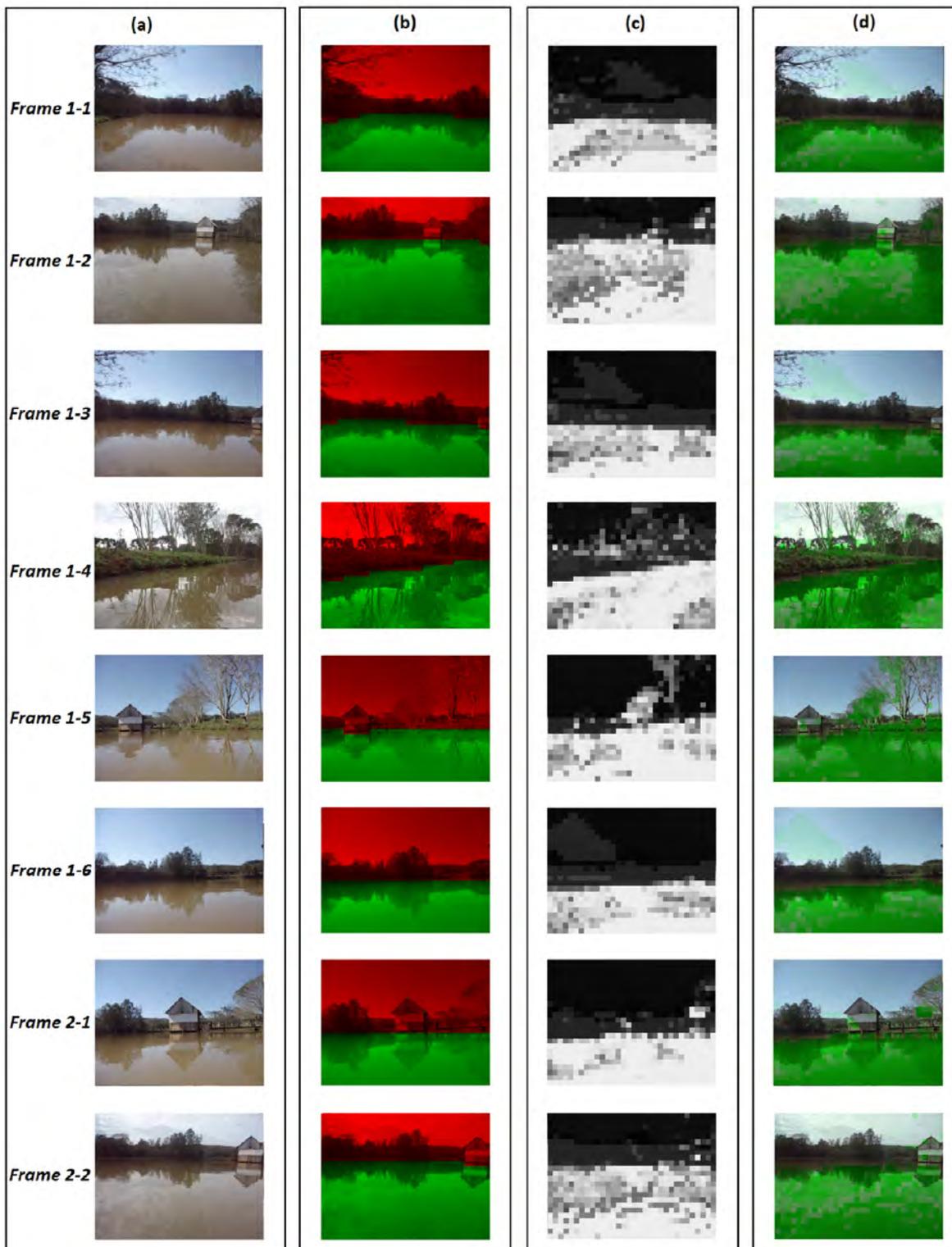
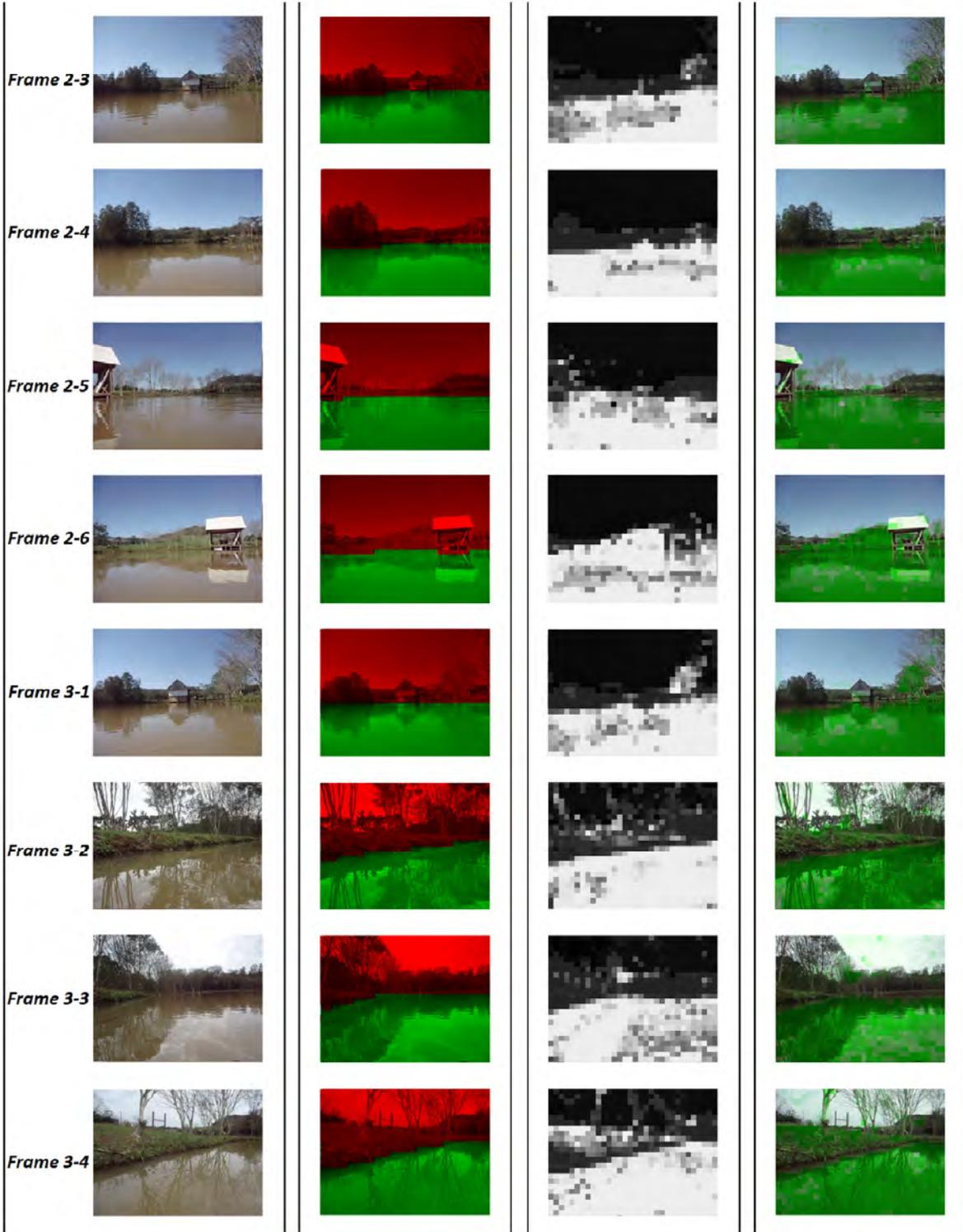


Figura D.1 – Conjunto de *frames* reavaliados utilizando o segundo classificador. Aonde (a) apresenta os *frames* de entrada; (b) Segmentação manual realizada; (c) O resultado obtido; E (d) União da classificação com a imagem de entrada.

APÊNDICE E – RESULTADOS OBTIDOS NA CLASSIFICAÇÃO DO SEGUNDO CENÁRIO





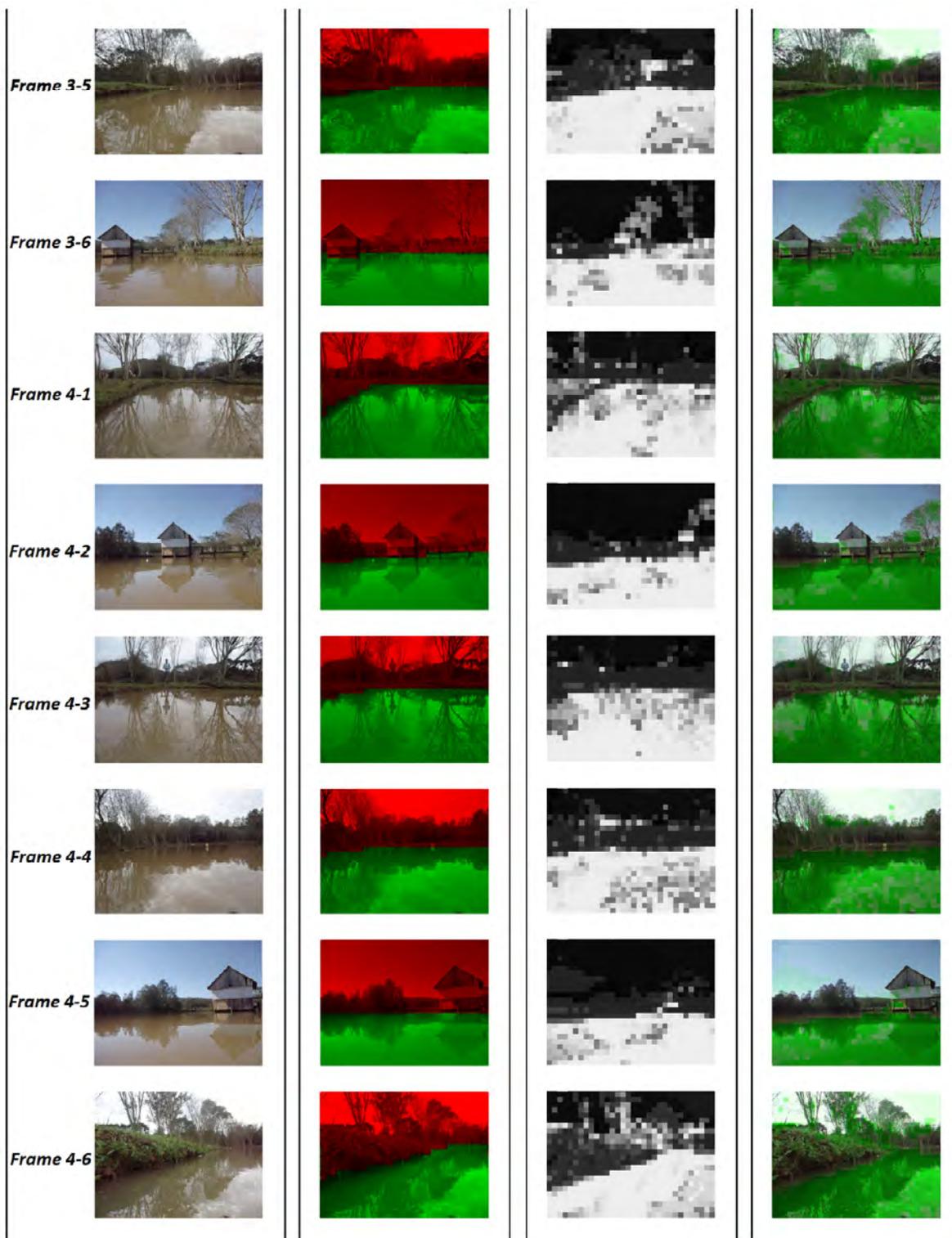


Figura E.1 – Resultados obtidos na classificação do segundo cenário avaliado em várias tomadas. As imagens em (a) apresentam os *frames* de entrada; (b) Segmentação manual realizada; (c) O resultado obtido; E (d) União da classificação com a imagem de entrada.