TIAGO ZIS

PROCESSO AUTOMATIZADO PARA EXTRAÇÃO DA LINHA DE CONTORNO DA NADADEIRA DORSAL DE CETÁCEOS EM IMAGENS DIGITAIS

Itajaí (SC), agosto de 2019

UNIVERSIDADE DO VALE DO ITAJAÍ

CURSO DE MESTRADO ACADÊMICO EM COMPUTAÇÃO APLICADA

PROCESSO AUTOMATIZADO PARA EXTRAÇÃO DA LINHA DE CONTORNO DA NADADEIRA DORSAL DE CETÁCEOS EM IMAGENS DIGITAIS

por

Tiago Zis

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Computação Aplicada.
Orientador: Rudimar Luís Scaranto Dazzi, Dr.
Co-Orientador: Andre Silva Barreto, Dr.

Itajaí (SC), agosto de 2019

FOLHA DE APROVAÇÃO

Esta página é reservada para inclusão da folha de assinaturas, a ser disponibilizada pela Secretaria do Curso para coleta da assinatura no ato da defesa.

AGRADECIMENTOS

Agradecimentos.

PROCESSO AUTOMATIZADO PARA EXTRAÇÃO DA LINHA DE CONTORNO DA NADADEIRA DORSAL DE CETÁCEOS EM IMAGENS DIGITAIS

Tiago Zis

Agosto / 2019

Orientador: Rudimar Luís Scaranto Dazzi, Dr

Co-Orientador: Andre Silva Barreto, Dr

Área de Concentração: Computação Aplicada

Linha de Pesquisa: Inteligência Aplicada

Palavras-chave: Biometria animal, foto identificação, visão computacional, detecção de objetos, segmentação semântica, algoritmo matting.

Número de páginas: 137

RESUMO

A identificação individual de organismos presentes na natureza é uma poderosa ferramenta de observação do comportamento animal, dinâmica demográfica e padrões de migração das espécies. No entanto, na maioria dos casos o trabalho de identificação individual pode ser muito dispendioso, devido ao constante processo de captura e recaptura de indivíduos para controle, através de marcadores artificiais, como por exemplo, colares e anilhas. Porém este conceito mudou a partir dos anos 1970 com a introdução da foto identificação, que possibilitou o reconhecimento de cada indivíduo de uma determinada espécie através de características únicas registradas nas imagens. No contexto da biologia marinha, existem trabalhos promissores que focam na identificação adotando técnicas de visão computacional e algoritmos de classificação baseados em inteligência artificial, para construir catálogos de imagens das populações do objeto de estudo. Contudo ainda existem vários obstáculos a serem ultrapassados, principalmente quando se trata de identificação individual de cetáceos, cuja a identificação é efetuada através da análise de marcas encontradas nas nadadeiras dorsais. Muitos trabalhos já foram desenvolvidos para solucionar o problema, no entanto poucos se aventuraram na construção de um processo automatizado de identificação do objeto de estudo. Assim sendo, este trabalho desenvolveu um processo automatizado para a etapa de extração da linha de contorno da dorsal de cetáceos, visando extinguir o processo de seleção manual de características do indivíduo. A criação deste processo, bem como a construção de uma ferramenta para a execução da tarefa foi dividida em três etapas: (i) localização e detecção de dorsais; (ii) segmentação da dorsal para destaca-la do contexto da cena; e (iii) extração da linha de contorno. Na primeira etapa utilizou-se a técnica de detecção de objetos com Redes Neurais Convolucionais SSD disponibilizada pela API do Tensorflow, cujos resultados da avaliação foram de AP 95,97%. A segunda etapa fez uso da técnica de segmentação semântica conhecida como DeepLab, que também apresentou resultados significativos ao atingir um valor mIoU de

70,3% para todas as classes envolvidas no processo. Já na etapa de extração das linhas de contornos, adotou-se a técnica de visão computacional conhecida como *matting*, dos seis algoritmos avaliados para esta tarefa, apenas um apresentou um comportamento atípico, os demais resultaram em uma precisão global acima de 82%, bem como um valor de *F-score* superior a 0,83.

AUTOMATED PROCESS FOR CONTOUR LINE EXTRACTION OF CETACEAN FIN IN DIGITAL IMAGES

Tiago Zis

August / 2019

Advisor: Rudimar Luís Scaranto Dazzi, Dr

Co-Advisor: Andre Silva Barreto, Dr

Area of Concentration: Applied Computer Science

Research Line: Applied Intelligence

Keywords: Animal biometric, photo identification, computer vision, object detection, semantic segmentation, matting algorithm.

Number of pages: 137

ABSTRACT

The individual identification of organisms in nature is a powerful tool for observing animal behavior, demographic dynamics and species migration patterns. However, in most cases individual identification can be very demanding due to the constant process of capturing and recapturing individuals for control, using artificial markers such as necklaces and rings. However, this concept changed from the 1970s with the introduction of photo identification, which allowed the recognition of individuals of a particular species through the presence unique characteristics recorded in images. In the context of marine biology, there are promising works that focus on identification by adopting computer vision techniques and classification algorithms based on artificial intelligence, to build image catalogs of the populations under study. However, there are still several obstacles to be overcome, especially when it comes to the individual identification of cetaceans, whose identification uses marks found on the dorsal fins. Much research has been done to solve the problem, but few have ventured to build an automated process for identifying the object of study. Therefore, this work developed an automated process for the extraction of cetacean's dorsal contour line, aiming to eliminate the process of manual selection of individual characteristics. The creation of this process, as well as the construction of a tool for the task was divided into three steps: (i) dorsal location and detection; (ii) dorsal segmentation to separate it from the context of the scene; and (iii) extraction of the contour line. In the first stage we used an object detection technique with SSD Convolutional Neural Networks provided by the Tensorflow API, with evaluation results of AP 95.97%. The second step made use of the semantic segmentation technique known as DeepLab, which also presented significant results reaching a value of mIoU 70.3% for all classes involved in the process. In the contour lines extraction stage, we adopted the computer vision technique known as matting. Of the six algorithms evaluated for this task, only one presented an atypical behavior, the others resulted in an overall accuracy above 82%, as well as as an F-score greater than 0.83.

LISTA DE ILUSTRAÇÕES

Figura 1: Exemplos de marcas únicas para identificação individual das espécies, (a) *Giraffa camelopardalis thornicrofti*, (b) *Equus quagga*, (c) *Acinonyx jubatus*, (d) *Megaptera novaeangliae*. 27
Figura 2: Exemplos de características de identificação, (a) pigmentação e (b) entalhes no contorno da dorsal. 28
Figura 3: Diagrama dos principais componentes presentes em um software de identificação individual. 30
Figura 4: Funções de ativação. (a) Sigmoide; (b) Tangente hiperbólica; (c) Unidade linear retificada. 32
Figura 5: Modelo de RNA. 33
Figura 6: Arquitetura básica de uma CNN. 34
Figura 7: CNN *kernel*. 34
Figura 8: Operação de *pooling*. 35
Figura 9: Arquitetura R-CNN. 36
Figura 10: Arquitetura *Fast* R-CNN. 37
Figura 11: Arquitetura *Faster* R-CNN. 37
Figura 12: Arquitetura SSD. 38
Figura 13: Caixas delimitadoras geradas pelas previsões dos *kernels* da rede. 39
Figura 14: Arquitetura R-FCN. 40
Figura 15: Classificação dos pixels da imagem. 41
Figura 16: Tipos de convoluções. (a) convolução dilatada; (b) convolução padrão. 42
Figura 17: Passos para criação do mapa de características com convolução dilatada. 43
Figura 18: Operação de múltiplas convoluções de dilatação. 44
Figura 19: Arquitetura do DeepLab. 45
Figura 20: Exemplo da operação de *matting*. 47
Figura 21: Avaliação da sobreposição das caixas delimitadoras para a detecção de objetos. 48
Figura 22: Correspondências dos pixels da linha de contorno. Em vermelho a linha do padrão verdade, em amarelo a linha predita pelo algoritmo. 51
Figura 23: Diagrama de execução do software Finscan. 55
Figura 24: DR, método manual de identificação individual 56
Figura 25: Notações primitivas e atributos de medidas do contorno da dorsal. 56
Figura 26: (a) notações primitivas do contorno da dorsal, (b) modelo de representação LSS e (c) modelo de representação HLS. 57
Figura 27: Resultados obtidos por Araabi et al. (2000), ao avaliar os algoritmos de correspondência das dorsais. 58
Figura 28: Passos para a etapa de geração do contorno da dorsal. 60
Figura 29: Gráfico de qualidade de segmentação, primeiro valor mínimo ideal para o *threshold* é t=65. 61
Figura 30: Pigmentação na dorsal dos golfinhos comuns (*Delphinus spp.*), encontrados em New Zealand. 62
Figura 31: Esquemas de subdivisão da dorsal, *grid* a esquerda e baseado no contorno a direita. 63
Figura 32: Conversão de curvatura da linha de contorno. (a) exemplo de linha do contorno de uma dorsal de golfinho, (b) conversão da linha de contorno em curvatura integral. 66
Figura 33: Modelo criado por Hughes e Burghardt (2016) para automatizar o processo de identificação individual de tubarão branco. 76

Figura 34: Diagrama do processo de detecção e extração da linha de contorno da dorsal para a ferramenta desenvolvida neste trabalho. 76
Figura 35: Precisão x tempo, cada forma geométrica representa a meta-arquitetura e as cores os extratores de características. 78
Figura 36: Exemplo de imagens que representam os critérios de exclusão definidos para o processo de seleção de imagens. 80
Figura 37: Exemplo de imagens que representam o segundo e terceiro critérios de exclusão. 81
Figura 38: Exemplo de imagens rotuladas para o treinamento de detecção de objetos. Rótulos: (a) animal, (b) animal metade e (c) animal parcial. 82
Figura 39: Interface do software VIA, para anotação de segmentos. 89
Figura 40: Gráfico da função de perda durante o treinamento com modelo pré-treinado. No eixo Y valor da função de perda, no eixo X número de passos do treinamento. 91
Figura 41: Gráfico da função de perda durante o treinamento sem o modelo pré-treinado. No eixo Y valor da função de perda, no eixo X número de passos do treinamento. 91
Figura 42: Recorte da dorsal para imagem original e o respectivo segmento, usando a caixa delimitadora obtida na etapa de detecção de objetos. 94
Figura 43: Passos para criação do *trimap*. (1) recorte do segmento na região da dorsal; (2) extração da linha de contorno do segmento; (3) criação da área de intersecção do *trimap* e sobreposição desta no semento da dorsal. 94
Figura 44: Exemplo dos resultados obtidos com os algoritmos *matting*. (a) *Learning Based*; (b) *Bayesian*; (c) *Knn*; (d) *Closed form*. 95
Figura 45: Exemplos dos resultados obtidos na etapa de extração da linha de contorno da dorsal. (a) *Learning Based*; (b) *Bayesian*; (c) *Knn*; (d) *Closed form*. 96
Figura 46: Gráfico com o número de acertos e erros encontrados durante a detecção de objetos, utilizado o IoU=0.5. 98
Figura 47: Exemplos de erros de detecção gerados pelo modelo SSD destacados em vermelho, em verde das detecções corretas para os objetos do escopo. (a) detectou o pneu como dorsal; (b) detectou o animal como dorsal; (c) detectou a nadadeira como dorsal. 100
Figura 48: Exemplos de detecção falso positivo gerados pelo modelo SSD destacados em amarelo, em verde as detecções corretas de dorsais. (a) detecção de dorsal parcialmente oclusa pela água; (b) detecção de dorsal não anotada no padrão verdade; (c) detecção de dorsal não anotado no padrão verdade devido ao seu tamanho e distância da câmera. 101
Figura 49: Exemplos de inconsistências geradas pela segmentação. (a) segmento extrapola a área da dorsal; (b) segmento inferior aos limites da dorsal. 102
Figura 50: Indivíduo cuja a dorsal não foi anexada a região segmentada. 103
Figura 51: Exemplo de sobreposição com a área de intersecção. (a) cobertura do segmento que extrapola a área da dorsal; (b) cobertura do segmento inferior aos limites da dorsal. 103
Figura 52: Resultados da etapa de extração da linha de contorno da dorsal, a esquerda a linha gerada e a direita a linha sobreposta a imagem da dorsal. 105
Figura 53: Gráfico de curvas para precisão e revocação dos algoritmos *matting* com limiar de corte da binarização 0,5. 107
Figura 54: Gráficos de curvas para precisão e revocação dos algoritmos *matting* com limiar de corte da binarização com a média ponderada. 108
Figura 55: Dorsais com linhas de contornos excedentes, a esquerda o trimap utilizado no processo de *matting* e a direita a linha resultante sobreposta a imagem original da dorsal. 109
Figura 56: Ilhas de segmentos indesejados gerados durante a da tarefa de binarização. 110
Figura 57: Imagens com resultado de precisão inferior a 0,5, as linhas brancas correspondem ao padrão verdade e as amarelas os resultados da extração da linha de contorno. (a) e (b) pixels

com pouco contraste foreground e background; (c) imagem tremida; (d) interferência do reflexo da luz; (e) erro na área de cobertura do trimap. 112

LISTA DE TABELAS

Tabela 1. Resultados obtidos para a etapa de identificação de indivíduos com base nos modelos escolhidos para a classificação de metadados. 69
Tabela 2. Resultados intermediários dos testes de desempenho para a detecção de objetos. 70
Tabela 3. Resultados dos testes de desempenho para a detecção da dorsal. 70
Tabela 4. Comparativo de acurácia e velocidade de processamento para as meta-arquiteturas de redes neurais da API de detecção de objetos, utilizando a base de dados COCO e caixas delimitadoras. 84
Tabela 5. Comparativo de tempo de processamento por passo em milissegundos. Primeiro os tempos obtidos pelos autores da API utilizando uma GPU, na sequência o tempo alcançado neste trabalho utilizando um CPU de 32 núcleos. 85
Tabela 6. Números relacionados ao processo de treinamento dos modelos pré-treinados escolhidos para este trabalho. 86
Tabela 7. Comparação de resultados obtidos entre o DeepLab v3+ e os demais modelos de alta performance, na base de dados de teste do PASCAL VOC 2012. 88
Tabela 8. Comparação de resultados obtidos entre o DeepLab v3+ e os demais modelos de alta performance, na base de dados de teste Cityscapes com anotações de contorno grosseiras. 88
Tabela 9. Comparação de resultados obtidos para os algoritmos *matting* no trabalho de Hughes e Burghardt (2015). 92
Tabela 10. Resultados obtidos durante a avaliação da etapa de detecção de objetos. 97
Tabela 11. Resultados obtidos para a avaliação da detecção de objetos com a métrica PASCAL VOC. 98
Tabela 12. Número de imagens inconsistentes recrutadas para o processo de *matting*, após a avaliação visual das configurações de espessura da área de intersecção do *trimap*. 104
Tabela 13. Resultados globais para cada combinação de algoritmo e limiar de corte da binarização. 106
Tabela 14. Levantamento quantitativo das imagens com resultado de precisão (PR) inferior e superior à 0,5. 111

LISTA DE ABREVIATURAS E SIGLAS

AP	Average Precision
API	Application Programming Interface
ASPP	Atrous Spatial Pyramid Pooling
CHW	Circular Haar Wavelet
CMYK	Cyan, Magenta, Yellow and Black
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DR	Dorsal Ratio
DTW	Dynamic Time-Warping
DWT	Discrete Wavelet Transform
GB	Gigabyte
GPS	Global Positioning System
GPU	Graphics Processing Unit
HLS	High-level String Representation
HOG	Histogram of Oriented Gradientes
Ifm	Information Flow Matting
IoU	Intersection Over Union
JSON	JavaScript Object Notation
LDA	Linear Discriminant Analysis
LIBGEO	Laboratório de Informática da Biodiversidade e Geomática
Lkm	Large Kernel Matting
LLS	Low-level String Representation
LNBNN	Local Naive Bayes Nearest Neighbor
LoG	Laplacian of Gaussian
LOOCV	Leaveone-out Cross Validation
mAP	Mean Average Precision
max	Máximo
MCA	Mestrado em Computação Aplicada
mIoU	Mean Intersection Over Union
PMC-BS	Projeto de Monitoramento de Cetáceos da Bacia de Santos
PMP-BS	Projeto de Monitoramento de Praias da Bacia de Santos
PNG	Portable Network Graphics
R-CNN	Regions of the Convolutional Neural Network
ReLU	Rectified Linear Unit
R-FCN	Region-based Fully Convolutional Networks
RGB	Red, Green and Blue
RNA	Redes Neurais Convolucionais
RoI	Regions of Interest
SIFT	Scale-invariant Feature Transform
SIMBA	Sistema de Informação de Monitoramento da Biota Aquática
SIMMAM	Sistema de Apoio ao Monitoramento de Mamiferos Marinhos
SSD	Single Shot Detector
SVM	Support Vector Machine
UNIVALI	Universidade do Vale do Itajaí
VIA	VGG Image Annotator

XML	Extensible Markup Language

LISTA DE SÍMBOLOS

∪	União
∩	Intersecção
∈	Pertence
Σ	Sigma
Δ	Delta

SUMÁRIO

1 INTRODUÇÃO 18
1.1 PROBLEMA DE PESQUISA 20
1.1.1 Solução Proposta 21
1.1.2 Delimitação de Escopo 22
1.1.3 Justificativa 22
1.2 OBJETIVOS 24
1.2.1 Objetivo Geral 24
1.2.2 Objetivos Específicos 24
1.3 METODOLOGIA 24
1.3.1 Metodologia da Pesquisa 24
1.3.2 Procedimentos Metodológicos 25
1.4 ESTRUTURA DA DISSERTAÇÃO 25
2 FUNDAMENTAÇÃO TEÓRICA 26
2.1 BIOMETRIA ANIMAL 26
2.1.1 Sistemas de reconhecimento de biometria animal 28
2.2 DETECÇÃO DE OBJETOS 30
2.2.1 Redes Neurais Artificiais (RNA) 31
2.2.2 CNN 34
2.2.3 *Regions of the Convolutional Neural Network* (R-CNN) 35
2.2.4 *Fast* R-CNN 36
2.2.5 *Faster* R-CNN 37
2.2.6 *Single Shot MultiBox Detector* (SSD) 38
2.2.7 *Region-based Fully Convolutional Networks* (R-FCN) 39
2.3 SEGMENTAÇÃO SEMÂNTICA 40
2.3.1 DeepLab 41
***2.4 MATTING* 45**
2.4.1 *Trimap* 46
2.5 MÉTRICAS DE AVALIAÇÃO 47
2.5.1 *Intersection Over Union* (IoU) 47
2.5.2 *Average Precision* (AP) 49
2.5.3 *F-score* 50
3 TRABALHOS RELACIONADOS 52
3.1 FINSCAN, UM SISTEMA DE IDENTIFICAÇÃO FOTOGRÁFICA PARA ANIMAIS MARINHOS 54
3.2 *THRESHOLD* NÃO SUPERVISIONADO PARA EXTRAÇÃO AUTOMÁRICA DA LINHA DE CONTORNO DA DORSAL DE GOLFINGOS ATRAVÉS DE FOTOGRAFIAS DIGITAIS NO SOFTWARE DARWIN 59

3.3 RECONHECIMENTO INDIVIDUAL DE GOLFINHOS UTILIZANDO A PIGMENTAÇÃO DA DORSAL 62
3.4 FOTO IDENTIFICAÇÃO DE BALEIA AZUL PARA DISPOSITIVOS MÓVEIS ATRAVÉS DA NADADEIRA DORSAL USANDO ALGORITMOS DE CLUSTERING E ESTIMATIVA DE COMPLEXIDADE LOCAL DAS CORES 64
3.5 REPRESENTAÇÃO DA CURVATURA INTEGRAL E ALGORITMOS DE CLASSIFICAÇÃO PARA IDENTIFICAÇÃO DE GOLFINHOS E BALEIAS 65
3.6 IDENTIFICAÇÃO DE CETACEOS UTILIZANDO METADADO 67
3.7 IDENTIFICAÇÃO INDIVIDUAL AUTOMATIZADA DE TUBARÕES BRANCOS 69
3.8 SOFTWARE SEMI-AUTOMATIZADO PARA IDENTIFICAÇÃO DE INDIVÍDUOS DA ESPÉCIE CARCHARODON CARCHARIAS ATRAVÉS DE FOTOGRAFIAS DA NADADEIRA DORSAL 71
3.9 ANÁLISE COMPARATIVA 72
3.10 CONSIDERAÇÕES 74
4 DESENVOLVIMENTO 75
4.1 DETEÇÃO DE OBJETOS 77
4.1.1 Base de dados 78
4.1.2 Seleção das imagens 79
4.1.3 Delimitação dos objetos de interesse 81
4.1.4 Modelo pré-treinado de rede neural 83
4.1.5 Treinamento 85
4.2 SEGMENTAÇÃO 86
4.2.1 Segmentação do objeto de interesse 87
4.3 EXTRAÇÃO DA LINHA DE CONTORNO DA DORSAL 92
5 RESULTADOS 97
5.1.1 Avaliação da etapa de detecção de objetos 97
5.1.2 Avaliação da etapa de segmentação 101
5.1.3 Análise visual da etapa de segmentação para criação do *trimap* 102
5.1.4 Avaliação da etapa de extração da linha de contorno da dorsal 104
6 CONCLUSÃO 113
6.1 CONTRIBUIÇÕES 115
6.2 SUGESTÕES PARA TRABALHOS FUTUROS 115
REFERÊNCIAS 117
GLOSSÁRIO 124
APÊNDICE A – ARQUIVO DE CONFIGURAÇÃO DO MODELO PRÉ-TREINADO ssd_resnet_50_fpn_coco 126

APÊNDICE B – ARQUIVO DE CONFIGURAÇÃO DO MODELO PRÉ-TREINADO rfcn_resnet101_coco..... 130
APÊNDICE C – ARQUIVO DE CONFIGURAÇÃO DO MODELO PRÉ-TREINADO faster_rcnn_nas 133
APÊNDICE D – CONFIGURAÇÕES PARA O TREINAMENTO DO DEEPLAB 136

1 INTRODUÇÃO

A identificação individual de organismos presentes na natureza é de grande valia para os biólogos. Este tipo de trabalho, permite explorar o comportamento de cada espécie, a dinâmica demográfica de grupos e os padrões de migração (BEARZI et al., 2005).

O método de identificação de indivíduos mais difundido no meio científico, exige a fixação de marcas artificiais como, por exemplo, etiquetas, anilhas, dispositivos de monitoramento via rádio ou *Global Positioning System* (GPS) (IRVINE; WELLS; SCOTT, 1982; OSBOURN et al., 2011; HOOVER et al., 2017).

Na maioria dos casos, esta técnica de identificação envolve o processo de captura e recaptura do indivíduo para fixação da marca ou observação de marcas existentes. Trabalhos como Saraux et al. (2011), Heide-Jørgensen et al. (2017) e Norman et al. (2018), evidenciam que o uso desta abordagem nos animais pode ocasionar estresse, mudanças no comportamento, ocorrência de infecções devido a perfurações e em alguns casos a morte de espécimes.

Em alguns grupos de animais, como os cetáceos (baleias e golfinhos), a captura não era um processo viável, fazendo com que os pesquisadores buscassem métodos alternativos para a identificação individual. Nos anos 1970 iniciou-se a técnica de identificação de cetáceos através de fotografias, mais difundida nas últimas décadas devido à popularização de máquinas fotográficas digitais (PERRIN; WÜRSIG; THEWISSEN, 2008). Essa técnica, tornou-se uma importante ferramenta no processo de identificação individual, em virtude de sua característica não invasiva de abordagem dos indivíduos do objeto de estudo.

Esta técnica não invasiva, faz uso de padrões de características presentes no corpo dos animais, durante o processo de identificação. Assim como as digitais de uma pessoa servem para identificá-la dentro de um grupo ou população, no reino animal é possível identificar um indivíduo de uma determinada espécie através de algum padrão de marcas (ARZOUMANIAN; HOLMBERG; NORMAN, 2005; CARTER et al., 2014; ZHELEZNIAKOV et al., 2015).

Em mamíferos marinhos da ordem dos cetáceos, as marcas de identificação, podem ser cortes nas bordas das nadadeiras dorsais dos golfinhos (MARKOWI; HARLIN; WURSIG, 2003),

coloração ou padrão da linha de contorno das nadadeiras das baleias jubarte (FRIDAY et al., 2000) ou calosidades na parte superior da cabeça das baleias francas (YURKOV; CHERNUKHA, 2015).

Com a difusão da técnica de identificação individual por meio de imagens digitais, houve um aumento no volume de dados gerados durante esse processo. Em vista disso, observou-se que o trabalho mecânico de avaliação visual das características de um indivíduo, em um catálogo de imagens, torna-se dispendioso e cansativo, podendo levar o pesquisador a cometer erros após longos períodos de trabalho.

Visando a redução do esforço necessário para esta atividade, pesquisadores das áreas de Ciências Biológicas e Ciência da Computação, juntaram esforços com intuito de criar softwares capazes de avaliar grandes volumes de dados, apresentando apenas os resultados relevantes encontrados durante o processo de comparação das características dos indivíduos. Permitindo dessa forma, maior precisão e agilidade na tomada de decisão do pesquisador.

Alguns exemplos de softwares que analisam o padrão das marcas nas imagens são: DARWIN (1993) que possibilita o usuário criar um catálogo de identificação de golfinhos através da avaliação de padrões das marcas existentes no contorno das nadadeiras dorsais; e a plataforma de softwares do Wildbook (Wildbook: Software to Combat Extinction, 2016), iniciativa que consiste em juntar vários pesquisadores do mundo, com intuito de criar algoritmos que possam auxiliar no processo de identificação de indivíduos de diferentes espécies de animais.

O conceito básico de funcionamento dos softwares de identificação individual se divide em duas etapas. A primeira, consiste em extrair as características de interesse do animal, aplicando técnicas de visão computacional sobre as imagens digitais, como a extração do contorno das dorsais de golfinhos (HALE, 2008) ou padrão de texturas da pelagem (ZHELEZNIAKOV et al., 2015). Na segunda, aplica-se algum tipo de algoritmo de análise e classificação de padrões nas características encontradas para o indivíduo avaliado, como por exemplo, *Dynamic Time-Warping* ou *Naive Bayesian Classifier*.

O processo de execução das duas etapas do software, resulta em um ranking dos indivíduos encontrados com as características de interesse similares ao indivíduo avaliado. Desse modo, o pesquisador poderá selecionar com maior precisão o indivíduo correspondente, incrementando as informações relacionadas a ele nos dados do catálogo de imagens do objeto de estudo.

Dentro do contexto apresentado, este trabalho focou na construção de um processo automatizado para extração das características de identificação das nadadeiras dorsais dos animais da ordem dos cetáceos, utilizando técnicas de detecção de objetos e segmentação semântica baseadas em Redes Neurais Convolucionais, que até o presente momento são consideradas como o estado da arte para este conjunto de técnicas de visão computacional. A construção deste processo permitiu a criação de uma ferramenta automatizada para extração da linha de contorno, que permite compartilhar os resultados obtidos com ferramentas que possam executar a etapa de identificação individual. O desenvolvimento deste trabalho também viabilizou a criação de um corpus de imagens de cetáceos com as devidas anotações de localização, segmentação e classificação dos indivíduos, para que possam ser utilizados em trabalhos futuros.

1.1 PROBLEMA DE PESQUISA

Apesar dos esforços relatados, sobre o desenvolvimento de softwares que auxiliam no processo de identificação individual de animais, ainda existem alguns obstáculos a serem transpassados.

Tratando-se de animais marinhos, mais especificamente os cetáceos, o principal obstáculo encontrado é a ausência de um mecanismo automatizado, que permita localizar o indivíduo na imagem e extrair das características necessárias para a identificação.

Este tipo de limitação pode ser constatado no software DARWIN, pois após a seleção da imagem do indivíduo que será identificado, o pesquisador é obrigado a informar manualmente a localização das extremidades inferiores de início e término da dorsal na imagem para delimitar a área de extração do contorno da dorsal.

Outro exemplo, é o trabalho de Weideman et al. (2017), onde o software desenvolvido demanda que o pesquisador realize o trabalho manual de recorte da imagem onde a dorsal do tubarão está localizada, antes de encaminha-la para a execução da identificação.

A principal justificativa apresentada pelos autores dos softwares citados, para a ausência de tal mecanismo, concentra-se no problema relacionado a qualidade das imagens obtidas pelos pesquisadores. Em imagens de animais marinhos, existem fatores presentes no habitat que podem

dificultar a automatização dos softwares de identificação individual, como por exemplo, reflexo da luz, ondas, esguichos de água e pouco contraste entre a coloração do indivíduo e do ambiente.

Contudo, iniciativas como o trabalho desenvolvido por Hughes e Burghardt (2016), para identificação de tubarões brancos através da nadadeira dorsal, bem como o trabalho de Yurkov e Chernukha (2015), identificação baleias franca usando as calosidades presente em suas cabeças. Ambos apresentaram bons resultados ao adotar técnicas hibridas que juntam algoritmos de visão computacional e aprendizado de máquina, para automatizar a primeira etapa de um software de identificação individual.

Apesar do bom desempenho apresentado nos resultados dos trabalhos citados, os autores não apresentam evidências que validem a eficiência das técnicas escolhidas para automatizar o processo de identificação individual, em imagens com condições distintas de iluminação, contraste e nitidez.

Portanto, este fato faz refletir sobre outro problema de pesquisa que deve ser explorado nesse trabalho. Mesmo sendo factível a possibilidade de replicação dos processos presentes nos trabalhos citados, para o desenvolvimento de um software de identificação de pequenos cetáceos. Será necessário buscar um meio de avaliar a eficiência deste software, aplicando testes em imagens que apresente condições adversas de ambiente.

Observando os fatos apresentados se faz pertinente o levantamento dos seguintes questionamentos para este projeto de dissertação:

- É possível implementar uma solução similar ao proposto no trabalho de Hughes e Burghardt (2016), em uma ferramenta de extração das características de identificação individual de cetáceos?
- De que maneira é possível avaliar a eficiência das técnicas adotadas na resolução do problema proposto, em imagens que apresentem condições adversas de ambiente, visando resultados qualitativos ou quantitativos?

1.1.1 Solução Proposta

Diversas abordagens utilizando técnicas de visão computacional já foram propostas para a solução do problema levantado (HALE, 2008; ANDREOTTI et al., 2017; CARVAJAL-GÁMEZ et al., 2017). No entanto, técnicas hibridas que misturam conceitos clássicos do processamento de

imagens digitais e algoritmos de aprendizado de máquina vem se destacando nos últimos quinze anos, principalmente na área de identificação individual de animais marinhos.

Portanto, este trabalho irá adotar como referência o trabalho de Hughes e Burghardt (2016), cujo objetivo foi a construção de uma ferramenta automatizada para identificação individual de grandes tubarões brancos.

Ao adotar os conceitos metodológicos propostos por Hughes e Burghardt (2016) no contexto de identificação de pequenos cetáceos, busca-se confirmar a seguinte hipótese:

h1: A implementação de algoritmos de visão computacional similares ao apresentado para tubarões brancos no trabalho de Hughes e Burghardt (2016), para a etapa de localização e extração do contorno da dorsal, também se aplica a cetáceos.

1.1.2 Delimitação de Escopo

Durante o levantamento dos trabalhos relacionados foi possível observar que em quase todos os casos, os pesquisadores focaram no desenvolvimento solução completa para o problema de identificação individual. Contudo, este trabalho focará apenas na automatização do processo de localização da nadadeira dorsal e extração da linha de contorno, em imagens digitais, bem como, na avaliação dos recursos incorporados nesse processo.

1.1.3 Justificativa

O Laboratório de Informática da Biodiversidade e Geomática (LIBGEO) da Univali executa várias atividades de pesquisa voltadas a espécies marinhas. Desde 2005 vem hospedando um sistema de gestão de dados sobre ocorrência de mamíferos marinhos, o Sistema de Monitoramento de Mamíferos Marinhos – SIMMAM (BARRETO et al., 2006), que atualmente possui mais de 30.000 registros de ocorrência armazenados em sua base de dados. O LIBGEO coordena o Projeto de Monitoramento de Praias da Bacia de Santos (PMP-BS), que é uma atividade desenvolvida para o atendimento da condicionante de licenciamento ambiental federal das atividades da Petrobras de produção e escoamento de petróleo e gás natural no Polo Pré-Sal da Bacia de Santos. O objetivo do projeto é avaliar o impacto de produção e escoamento de petróleo sobre as aves, tartarugas e mamíferos marinhos, através do monitoramento das praias e do atendimento veterinário aos animais

debilitados e coleta dos mortos. Uma das atribuições do LIBGEO é a gestão dos dados destas ocorrências de animais na área do projeto.

Um dos pontos importantes para avaliar o possível impacto é conhecer a distribuição dos organismos e como estas podem variar ao longo do tempo. Um dos modos de se realizar esta avaliação é o desenvolvimento de modelos de distribuição das espécies encontradas durante as atividades de monitoramento, predição dos possíveis motivos de ocorrência de encalhes correlacionando as variáveis ambientais envolvidas e as ocorrências de interações antrópicas evidenciados no momento do atendimento das ocorrências.

Portanto, uma forma de contribuir com a construção de um modelo de distribuição de espécies e que vem de encontro à proposta deste trabalho, trata-se de melhorar o desempenho do trabalho dos pesquisadores na atividade de identificação individual de pequenos cetáceos, buscando a construção de um software consistente que automatize a etapa de localização e extração das características da dorsal de cada indivíduo, encontrados durante as atividades de monitoramento do PMP-BS. A identificação destes indivíduos utilizando imagens do próprio PMP-BS em conjunto com outras bases de dados de imagens de projetos como, as imagens obtidas durante a execução do Projeto de Monitoramento de Cetáceos da Bacia de Santos (PMC-BS) (SISPMC, 2016), permitirá avaliar a distribuição espacial destes indivíduos na Bacia de Santos e consequentemente servirá como uma ferramenta que contribuirá com a avaliação do impacto das atividades de produção e escoamento de petróleo e gás sobre estes animais.

Adicionalmente, este trabalho também terá um papel importante nas pesquisas que envolvem o uso de algoritmos de visão computacional, tendo em vista que os métodos apresentados até o momento desconsideram a variabilidade da qualidade das imagens obtidas, bem como o envolvimento de condições ambientais adversas que interferem na visualização do objeto do estudo.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Construir um processo para automatização da etapa de extração das características de identificação das nadadeiras dorsais para cetáceos, através de técnicas de visão computacional.

1.2.2 Objetivos Específicos

1. Implementar e avaliar um método para detecção e extração de nadadeiras dorsais em imagens de cetáceos.
2. Implementar e avaliar uma técnica de extração da linha de contorno da nadadeira dorsal
3. Desenvolver uma ferramenta que entregue as linhas de contornos das dorsais extraídas de imagens digitais em arquivos do formato Portable Network Graphics (PNG), para que possam ser utilizados em um software de identificação individual de cetáceos.

1.3 METODOLOGIA

Nesta seção, serão apresentados os procedimentos metodológicos que conduziram a pesquisa, bem como as atividades executadas para cumprir os objetivos do trabalho.

1.3.1 Metodologia da Pesquisa

Este trabalho adotará o método indutivo, tendo em vista a necessidade de confirmar a hipótese levantada para o problema proposto.

A pesquisa terá abordagem quantitativa, pois pretende avaliar os resultados obtidos durante os testes de desempenho da solução proposta para o trabalho, através do uso de métricas adotadas pelos pesquisadores da área de visão computacional.

Quanto a natureza da pesquisa, pode-se considerar que esta pesquisa é aplicada, pois tem por objetivo aplicar na prática uma solução para o problema proposto, considerando o interesse do LIBGEO em ter um produto que contribua com os temas de pesquisas realizados no laboratório.

1.3.2 Procedimentos Metodológicos

Do ponto de vista metodológico, foi efetuada uma pesquisa bibliográfica através do processo de revisão sistemática, para definir o estado da arte das aplicações que empregam técnicas de visão computacional na extração das características biométricas das nadadeiras dorsais, com intuito de utiliza-las na identificação individual de animais marinhos.

1.4 ESTRUTURA DA DISSERTAÇÃO

O trabalho está dividido em seis capítulos. O Capítulo 1 contextualiza do tema proposto para o trabalho.

O Capítulo 2 aborda a fundamentação teórica para compreender melhor o contexto da proposta deste trabalho, passando pelos conceitos de biometria animal e as principais técnicas de visão computacional utilizadas neste trabalho, detecção de objetos, segmentação de imagens e extração da linha de contorno.

No Capítulo 3 é apresentado os resultados obtidos da pesquisa efetuada, buscando levantar as informações relacionadas ao estado da arte do processo de obtenção das características de identificação dos animais marinhos da ordem dos cetáceos. Durante a pesquisa abordou-se alguns conceitos voltados à revisão sistemática da literatura.

O Capítulo 4 descreve as etapas da construção do processo de automatização da etapa de extração das características de identificação das nadadeiras dorsais.

Tanto a avalição de cada etapa desenvolvida no Capitulo 4, bem como as discussões destas estão presentes no Capítulo 5.

Já o Capítulo 6 é destinado ao fechamento do tralho e contempla as conclusões e sugestões para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresentará alguns conceitos teóricos que embasam as pesquisas realizadas para o desenvolvimento deste trabalho. Entre os assuntos apresentados estão, a definição de biometria animal e os conceitos de detecção de objetos, segmentação, algoritmos *matting* e métricas de avaliação.

2.1 BIOMETRIA ANIMAL

Trata-se de um campo de pesquisa em constante evolução, onde os pesquisadores das áreas de ciências biológicas e ecologistas avaliam os padrões de características dos animais visando a classificação das espécies e identificação de indivíduos (KUMAR et al., 2017).

Uma das técnicas mais populares neste contexto de estudo é conhecida como foto identificação (SPEED; MEEKAN; BRADSHAW, 2007). Esta técnica consiste na aquisição de fotografias dos animais do objeto de estudo, buscando descrever as características relevantes que permitam a identificação dos indivíduos, através da observação das particularidades morfológicas e comportamentais de cada espécie (KUMAR et al., 2017).

A identificação dos indivíduos através da observação das características biométricas dos mesmos, é considerada um recurso importante em estudos relacionados ao comportamento animal, controle populacional e rastreamento individual ao longo do tempo (PERRIN; WÜRSIG; THEWISSEN, 2008).

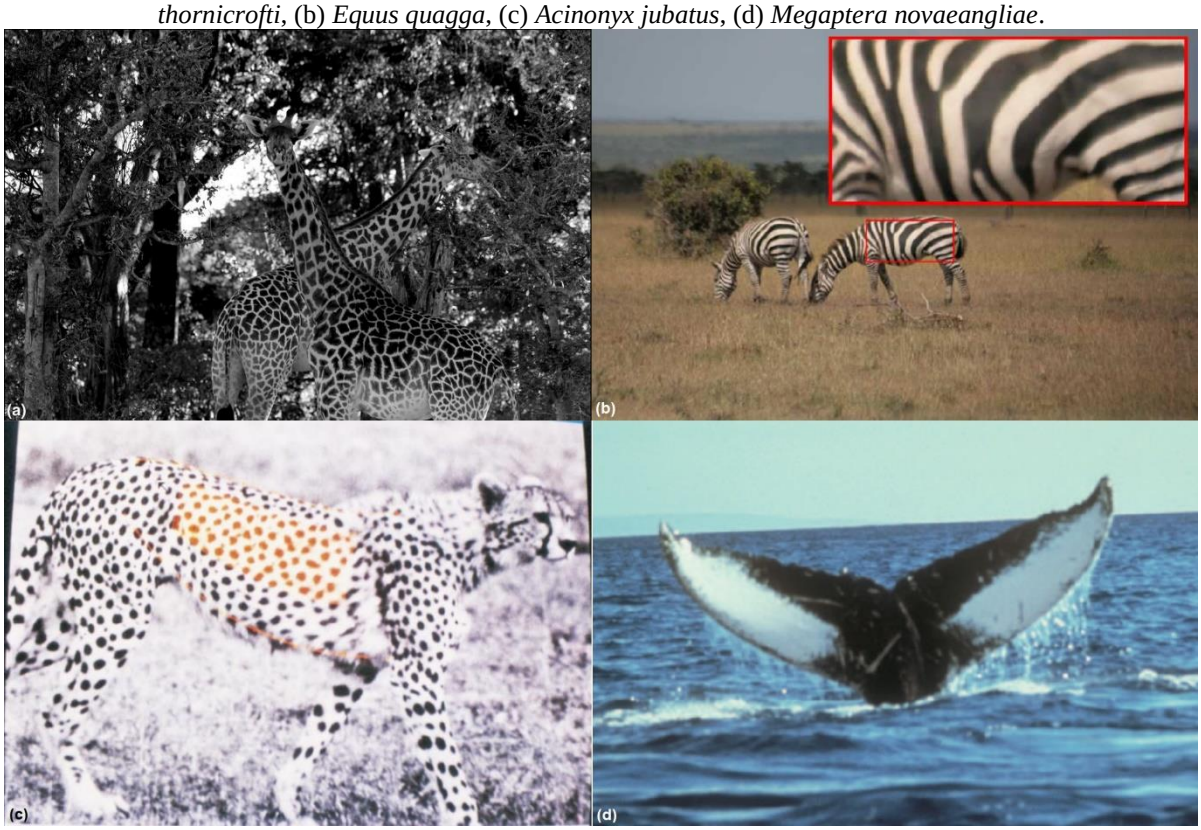
O levantamento dos padrões morfológicos e comportamentais dos animais conduzidos pelos pesquisadores, consiste em capturar informações visuais de diferentes ângulos e fontes, aplicando métodos de amostragem conhecidos como captura e recaptura de marcas (SPEED; MEEKAN; BRADSHAW, 2007).

Alguns exemplos de marcas avaliadas que pode-se citar são:

- Os padrões de listras em zebras *Equus quagga* (LAHIRI et al., 2011) (Figura 1b);

- Os pontos presentes na pelagem dos guepardos *Acinonyx jubatus* (KELLY, 2001) e girafas *Giraffa camelopardalis thornicrofti* (HALLORAN; MURDOCH; BECKER, 2014) (Figura 1a e Figura 1c); e
- A pigmentação aparente na cauda das baleias jubarte *Megaptera novaeangliae* (TITOVA et al., 2018) (Figura 1d).

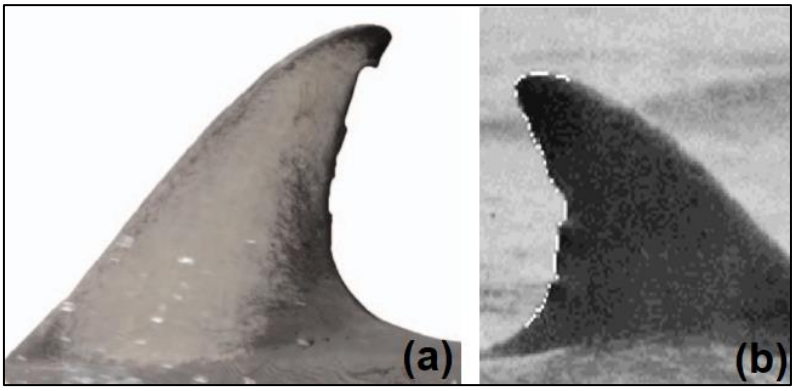
Figura 1: Exemplos de marcas únicas para identificação individual das espécies, (a) *Giraffa camelopardalis thornicrofti*, (b) *Equus quagga*, (c) *Acinonyx jubatus*, (d) *Megaptera novaeangliae*.



Fonte: Adaptado de Halloran, Murdoch e Becker (2014); Lahiri et al. (2011); Kelly (2001); Perrin, Würsig e Thewissen (2008).

No caso de cetáceos, as características de identificação mais evidentes são, a pigmentação da nadadeira dorsal ou caudal (GILMAN et al., 2016) (Figura 2a), e os entalhes no entorno das dorsais gerados por interações ambientais ou antrópicas (KREHO et al., 1997) (Figura 2b), sendo a segunda, o tipo de marca natural adotada com mais frequência por pesquisadores no processo de identificação individual de golfinhos, uma vez que nem todas as espécies destes tipos de animais apresentam pigmentação nas dorsais.

Figura 2: Exemplos de características de identificação, (a) pigmentação e (b) entalhes no contorno da dorsal.



Fonte: Adaptado de Gilman et al. (2016); Kreho et al. (1997).

Assim como a impressão digital serve para identificar os seres humanos, o modelo biométrico baseado nos entalhes no entorno das dorsais destes animais, funciona como um padrão de identificação que permite distinguir cada indivíduo em um grupo ou população. Este tipo de padrão é observado e fotografado pelos pesquisadores quando os animais expõem as dorsais para fora da água durante o seu ciclo respiratório.

Além disso, este tipo de abordagem de identificação possui duas vantagens importantes. A primeira é a permanência a longo prazo, pois apesar de ocorrer a cicatrização dos entalhes nas dorsais, as marcas ficarão presentes ao longo de toda a vida do animal, ou seja, o local afetado não se regenera. Já a segunda, refere-se ao fato de que as marcas podem ser vistas mesmo que a dorsal do indivíduo esteja posicionada para a direita ou esquerda (PERRIN; WÜRSIG; THEWISSEN, 2008).

2.1.1 Sistemas de reconhecimento de biometria animal

Os sistemas especializados em reconhecimento de biometria animal são também conhecidos e categorizados como sistemas de classificação de espécies ou identificação individual. Tem por objetivo a detecção e classificação de animais através do reconhecimento dos padrões de características (PERRIN; WÜRSIG; THEWISSEN, 2008).

Apesar de existir sistemas cujo objetivo é a classificação de animais de diferentes espécies, como por exemplo o iNaturalist (2018). A maioria dos sistemas disponíveis para identificação

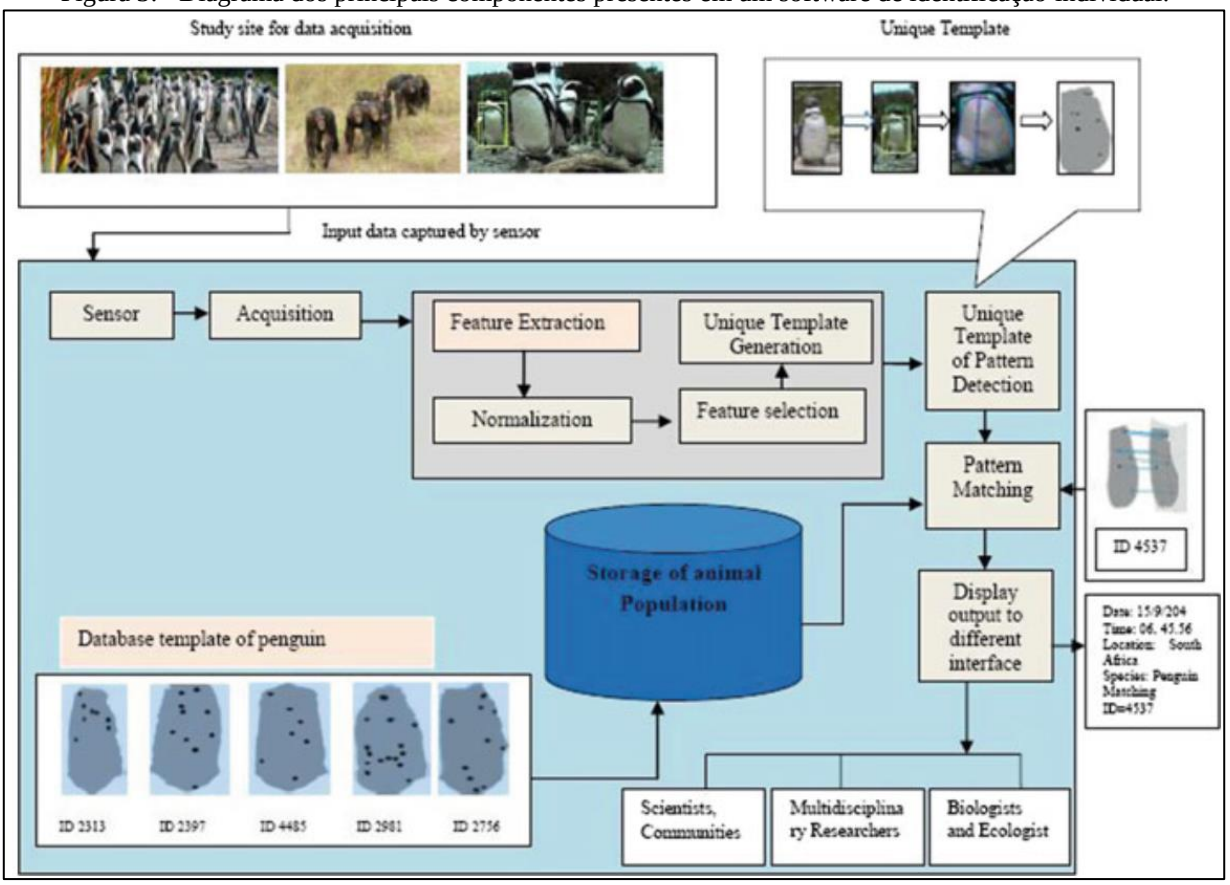
individual focam na identificação de uma única espécie como o caso do Wildbook for Whale Sharks (2018), voltado a identificação de tubarões baleia.

Basicamente, estes sistemas fazem uso de técnicas de visão computacional como meio de detecção, aquisição e representação computacional das características morfológicas e biométricas dos animais. Os dados coletados são extraídos e categorizados para a geração de modelos morfológicos da espécie, onde posteriormente são processados por algoritmos desenvolvidos para a etapa de identificação individual, também conhecida como etapa de comparação de indivíduos ou do termo em inglês *matching*.

Kumar et al. (2017), descreve que um software consistente para identificação individual baseado em biometria animal, conta com seis componentes importantes (Figura 3):

1. Sensores: equipamentos utilizados para aquisição de dados, por exemplo, máquinas fotográficas ou câmeras de armadilhas fotográficas (*trapping camera*).
2. Detecção da espécie: utilizando como base as características morfológicas e biométricas dos animais.
3. Armazenamento: capacidade de armazenar os dados coletados e processados.
4. *Matching*: análise de correspondência de similaridade das imagens consultadas em relação as imagens armazenadas no banco de dados, ou seja, execução do processo de comparação e identificação dos indivíduos.
5. *Ranking*: classificação dos resultados encontrados no processo de identificação individual, através da delimitação de um valor de corte dos resultados.
6. Apresentação: visualização dos resultados obtidos.

Figura 3: Diagrama dos principais componentes presentes em um software de identificação individual.



Fonte: Kumar et al. (2017).

2.2 DETECÇÃO DE OBJETOS

A detecção de objetos é uma das técnicas de visão computacional cujo o objetivo é determinar onde os objetos estão localizados em uma imagem e definir a qual categoria cada objeto pertence. Conforme Zhao et al. (2018), nos últimos anos tanto na área de pesquisa quanto no desenvolvimento de aplicações o uso desta técnica foi impulsionado pela adoção de recursos de aprendizado de máquina, como por exemplo, as Redes Neurais Artificiais (RNA). As RNA permitiram melhorar o desempenho dos algoritmos de detecção ao proporcionar recursos capazes de entender a complexidade da dinâmica dos objetos na cena.

Zhao et al. (2018) descreve que a detecção de objetos é basicamente composta por três passos:

1. Seleção de região informativa: consiste em utilizar uma técnica de busca de objetos em qualquer região da imagem, como por exemplo, a janela deslizante multi-escala;

2. Extração de características: técnica de reconhecimento de objetos, que utiliza descritores de características como, SIFT[1], HOG[2] e *Haar-like*; e
3. Identificação: identifica o objeto entre as distintas classes de um modelo, alguns exemplos de técnicas implementadas neste passo são, *Support Vector Machine* (SVM) e *AdaBoost*.

Apesar da forma simplista que foi descrita a técnica de detecção de objetos, os modelos adotados neste trabalho utilizam alguns recursos sofisticados para melhorar o desempenho da tarefa, como por exemplo, as redes neurais convolucionais do termo inglês *Convolutional Neural Network* (CNN), descritores de região de interesse e classificadores de objetos. Contudo antes de descrever sobre as principais características para cada arquitetura de detector de objetos, será necessário contextualizar alguns elementos que constituem uma RNA e também descrever os componentes necessários para transforma-la em uma CNN.

2.2.1 Redes Neurais Artificiais (RNA)

Nielsen (2015) descreve a RNA como um paradigma de programação inspirado biologicamente no cérebro humano, que permite um computador aprender a partir de um conjunto de dados. Ou seja, a RNA consiste em uma coleção de *perceptrons* (neurônios), que estão conectados através de unidades de camadas ocultas e por sua vez são ativados através de funções de ativação, uma representação análoga das sinapses neurais.

As funções de ativação possuem um papel importante na rede neural, elas evitam que esta transforme-se em um modelo linear, ao decidir quando um *perceptron* deve ou não ser ativado (SHANMUGAMANI, 2018).

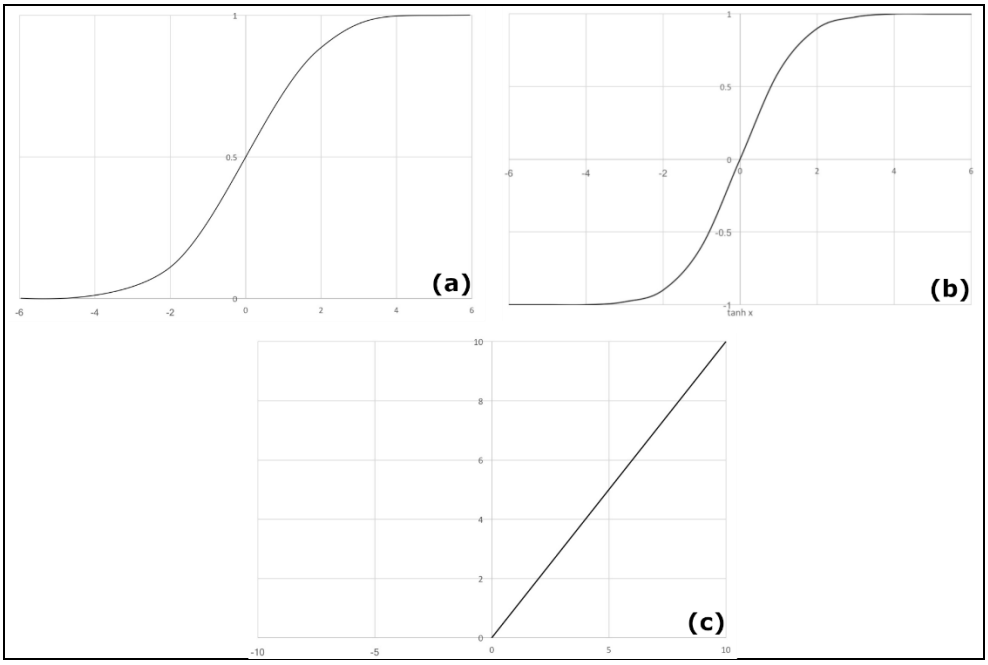
Algumas das funções mais utilizadas são:

- Sigmoide: pode transformar valores em probabilidades, e também pode ser utilizada em classificações binárias (Figura 4 (a));

[1] Scale-invariant feature transform (SIFT) ou do português, Transformação de recurso invariante de escala.
[2] Histogram of oriented gradientes (HOG) termo em inglês para Histograma de gradientes orientados.

- Tangente hiperbólica: semelhante a sigmoide permite suavizar e diferenciar os valores, porém é mais estável (Figura 4b); e
- Unidade linear retificada[3]: gera esparsidade entre os neurônios da rede uma vez que pode deixar passar apenas os valores maiores, isto gera o desuso de alguns neurônios na rede (Figura 4c).

Figura 4: Funções de ativação. (a) Sigmoide; (b) Tangente hiperbólica; (c) Unidade linear retificada.

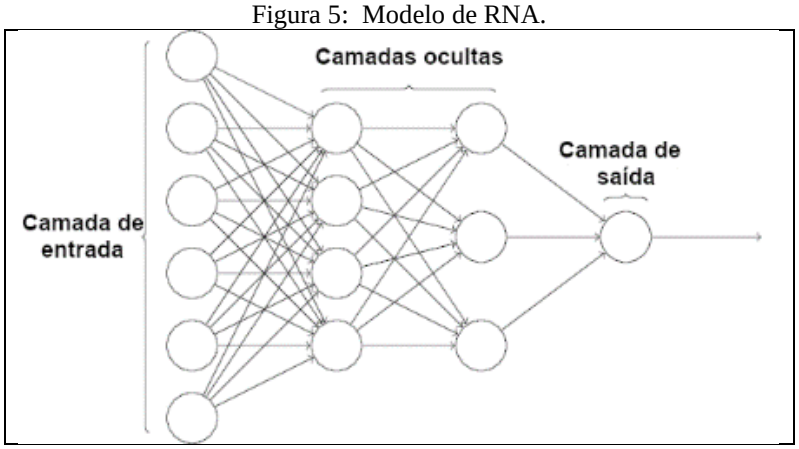


Fonte: Adaptado de Shanmugamani (2018).

Um modelo básico de RNA pode ser definido conforme a Figura 5, neste caso a primeira camada é a de entrada de dados, as camadas do meio ou camadas ocultas formam a base não linear que mapeia as camadas de entrada para última camada, a de saída. Os modelos de aprendizados de uma rede são gerados a partir do cálculo ponderado dos pesos e vieses, e estes são atualizados a cada passo do treino através do cálculo de uma função de perda utilizando os dados do padrão verdade como referência (SHANMUGAMANI, 2018).

[3] Tradução em português para o termo em inglês *Rectified Linear Unit* (ReLU)

Figura 5: Modelo de RNA.



Fonte: Adaptado de Nielsen (2015).

A função de perda é de extrema importância na construção de um modelo de RNA, pois está diretamente ligada à camada de saída de uma rede neural e consequentemente calcula o erro gerado pelo modelo ao produzir um valor de saída (NIELSEN, 2015). Outra funcionalidade que pode ser atribuída a função de perda está relacionada a observação dos valores retornados durante o treinamento para definir se o mesmo deve ou não ser encerrado, ou seja, o encerramento poderá ser efetuado ao observar que o erro não pode ser reduzido ou que o valor não apresenta uma variação significativa.

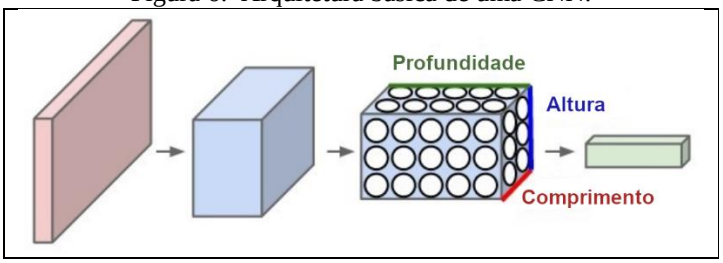
Em problemas em que o resultado final do modelo deve gerar uma classificação dos dados de entrada, a RNA deve contar com uma função de ativação na camada de saída, que permita classificar corretamente as informações. A função comumente utilizada nestes casos é a *Softmax*, que converte todos os valores de saída em probabilidades de pertencerem a uma determinada classe do modelo, ou seja, divide cada valor pela soma dos demais para criar o grau de confiança para a classificação (NIELSEN, 2015).

Alguns modelos de rede neural tendem a ter problemas de sobreajuste, também conhecido pelo termo em inglês *overfitting*. Trata-se dos casos onde o modelo adapta-se aos dados de treinamento, porém com a entrada de novos valores gera vários erros nos resultados (NIELSEN, 2015). Para resolver este problema as RNA contam com métodos de regularização, como o *Dropout* que remove aleatoriamente algumas unidades das camadas da rede, o L1 penaliza os valores absolutos dos pesos tendendo a zera-los e o L2 que penaliza os valores quadrados dos pesos para reduzi-los durante o treinamento (SHANMUGAMANI, 2018).

2.2.2 CNN

As redes convolucionais essencialmente são muito parecidas com as RNA, porém não possuem os neurônios totalmente conectados. Os neurônios de uma CNN são organizados volumetricamente, transformando um determinado volume de dados de entrada em um volume diferente de dados de saída (SHANMUGAMANI, 2018). Em aplicações voltadas a visão computacional, este volume de dados pode ser representado pelas camadas de cores RGB da imagem (Figura 6).

Figura 6: Arquitetura básica de uma CNN.

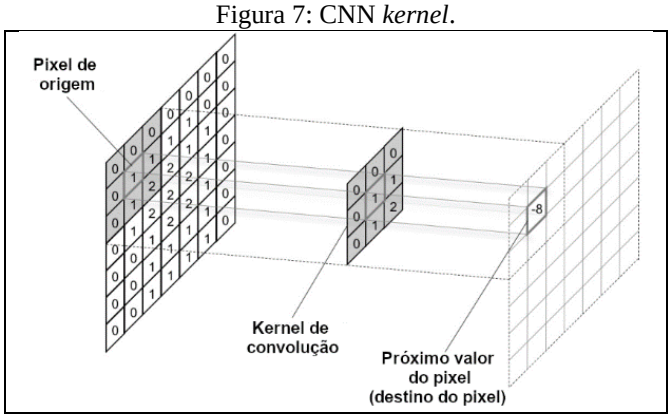


Fonte: Adaptado de Shanmugamani (2018).

2.2.2.1 *Kernel*

A operação de convolução de uma CNN funciona como extrator de características de uma imagem, preservando o relacionamento entre um conjunto de pixels ao aplicar um determinado filtro com uma configuração de *kernel*. O *kernel* é formado por dois parâmetros, o primeiro diz respeito ao tamanho do mesmo (e.g. 3x3), já o segundo parâmetro remete ao número de passos executados durante deslocamento (SHANMUGAMANI, 2018), a Figura 7 demostra um exemplo de funcionamento do *kernel*.

Figura 7: CNN *kernel*.



Fonte: Adaptado de Shanmugamani (2018).

2.2.2.2 *Pooling*

As camadas de *pooling* são inseridas entre as camadas convolucionais, com intuito de reduzir a dimensão dos dados para acelerar o processamento da informação e também é utilizada como uma técnica de regularização para evitar o *overfitting*. As operações mais comuns para esta camada consistem em obter o valor máximo (*max-pooling*) ou o valor médio para cada conjunto de pixels (SHANMUGAMANI, 2018). A Figura 8 demonstra um exemplo das operações para a camada.

Figura 8: Operação de *pooling*.

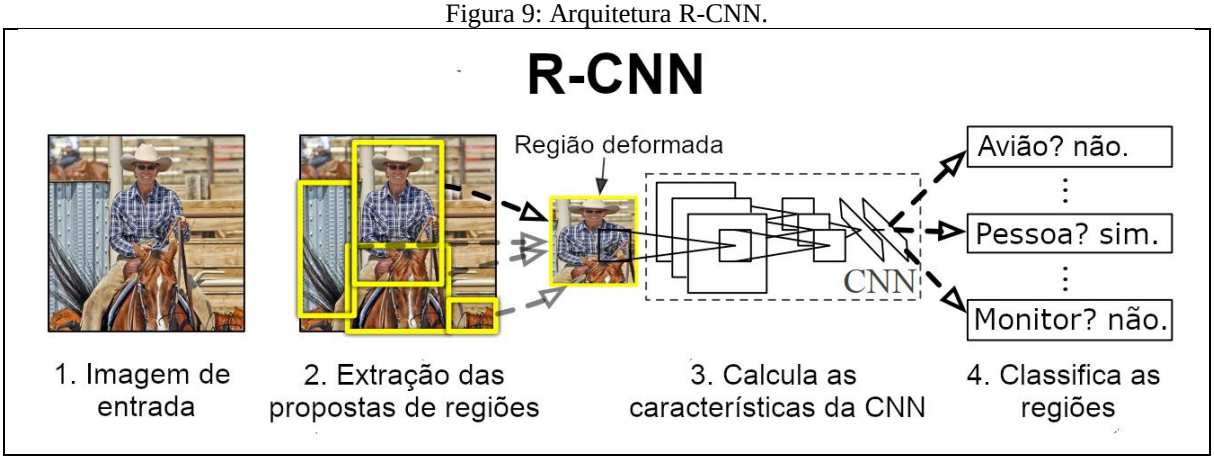
Fonte: Adaptado de Shanmugamani (2018).

2.2.3 *Regions of the Convolutional Neural Network* (R-CNN)

Este modelo de detecção de objetos é o precursor da família R-CNN, bem como foi o primeiro a utilizar a busca seletiva descrita por Uijlings et al. (2013), para criar algumas propostas de regiões de interesse[4] (SHANMUGAMANI, 2018). A arquitetura do modelo está representada na Figura 9.

[4] Regiões de interesse termo em português para *Regions of Interest* (RoI).

Figura 9: Arquitetura R-CNN.



Fonte: Adaptado de Girshick et al. (2014).

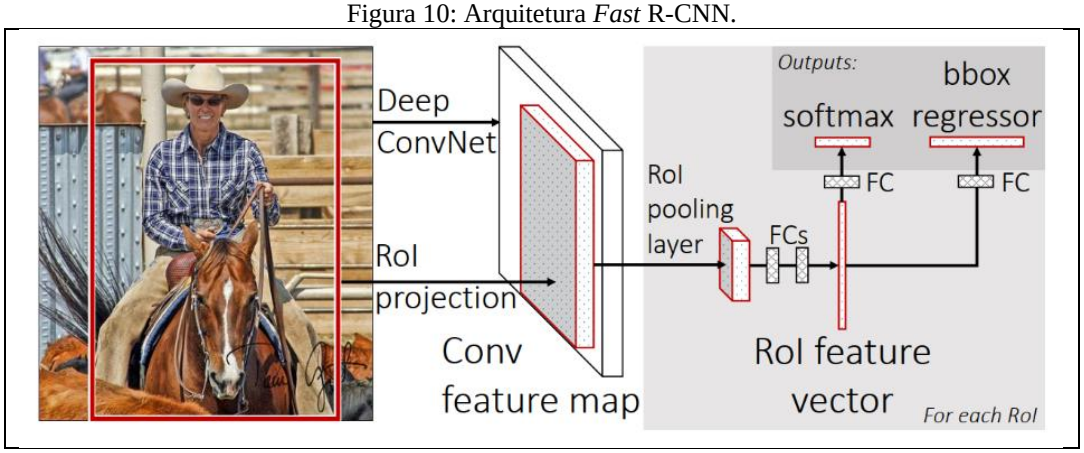
Conforme Girshick et al. (2014), 2000 propostas de regiões são extraídas a partir de uma imagem de entrada, cada região é redimensionada para um tamanho fixo e processada por uma CNN com intuito de obter os mapas de características de identificação do objeto. No final estes mapas fazem uso de uma SVM linear para classificar o objeto a partir das características obtidas na etapa anterior.

Shanmugamani (2018) aponta três fatores que trazem desvantagens ao modelo proposto. O primeiro é que o número de regiões propostas a serem processadas é muito grande, tornando a tarefa de detecção lenta. Outro fator negativo é a presença de três classificadores que precisam ser treinados e isto aumenta o número de parâmetros da rede. O último é a ausência de um treinamento ponta a ponta.

2.2.4 *Fast* R-CNN

Na nova versão do modelo desenvolvido por Girshick (2015), dado uma imagem de entrada e um conjunto de regiões de interesse o processo de identificação passa por uma rede totalmente convolucional, para posteriormente extrair os mapas de características fixos através de uma camada de *max-pooling* para cada região de interesse. Finalizando o processo em camadas totalmente conectadas que produzem na saída um vetor de probabilidades gerados pelo *softmax* e outro vetor de caixas delimitadoras de regressão. A Figura 10 descreve a arquitetura para esta atualização do modelo. Girshick (2015) descreve em seu trabalho que esta alteração do novo modelo deixou o processo 9 vezes mais rápido que o seu antecessor.

Figura 10: Arquitetura *Fast* R-CNN.

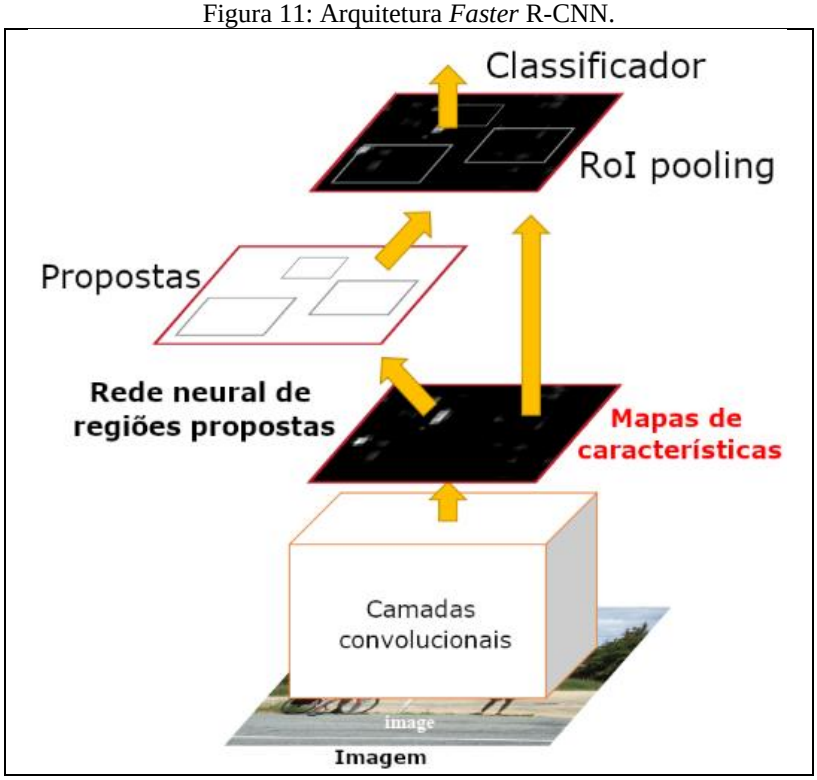


Fonte: Girshick (2015).

2.2.5 *Faster* R-CNN

Este modelo foi o último da família R-CNN a ser desenvolvido, e também é um dos modelos adotados neste trabalho. A Figura 11 apresenta a arquitetura do modelo com o ajuste efetuado para o mesmo.

Figura 11: Arquitetura *Faster* R-CNN.



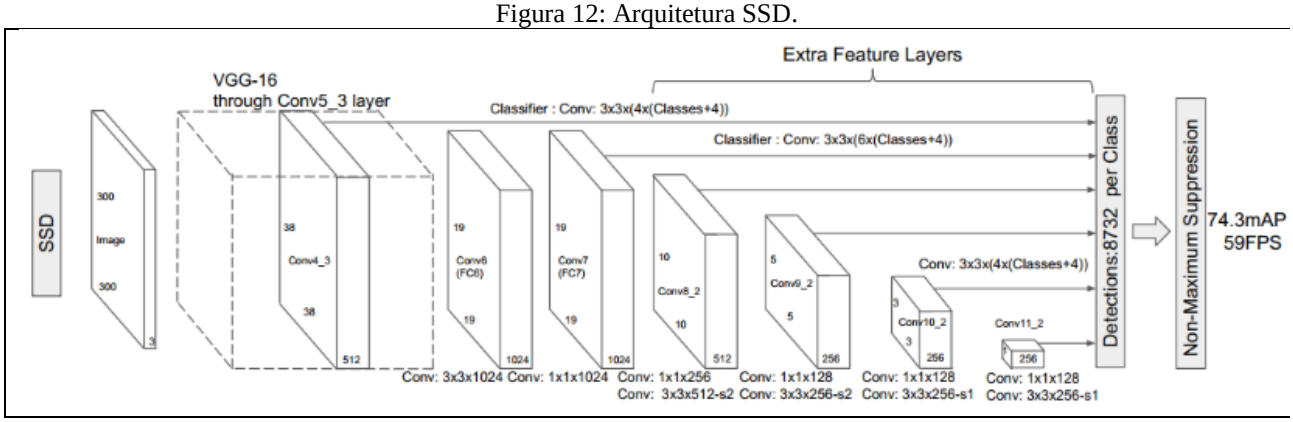
Fonte: Adaptado de Ren et al. (2015).

Ren et al. (2015), descrevem que a principal alteração deste modelo em relação aos demais, foi a substituição do algoritmo gerador de regiões propostas, por uma rede totalmente convolucional que cria um mapa de características para gerar um conjunto de regiões propostas com objetos retangulares, e cada um com sua respectiva pontuação de objetividade.

2.2.6 *Single Shot MultiBox Detector* (SSD)

Enquanto os modelos da família R-CNN apresentam uma arquitetura que utiliza mais de uma rede para obter a localização dos objetos. O modelo SSD desenvolveu uma arquitetura de rede unificada, para prever a localização de múltiplos objetos e descrever as caixas delimitadoras dos mesmos, bem como entrega uma rede de alta performance capaz de rodar a 22 FPS com imagens de resolução 500x500 pixels (SHANMUGAMANI, 2018). A Figura 12 apresenta o desenho da arquitetura criada para este modelo de rede.

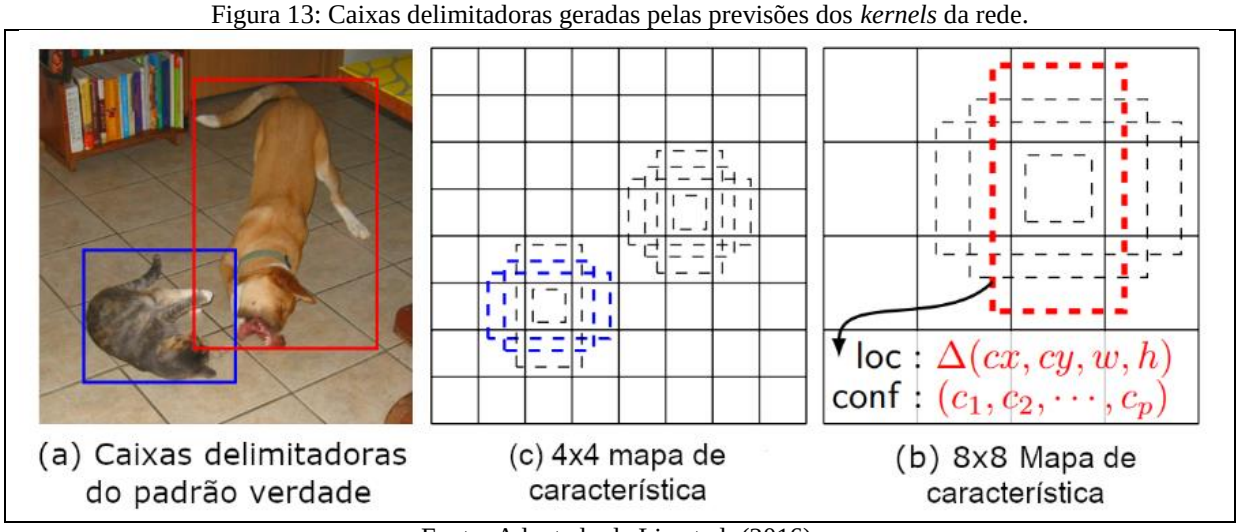
Figura 12: Arquitetura SSD.



Fonte: Liu et al. (2016).

Conforme pode-se observar na Figura 12, cada etapa do modelo convolucional representa uma camada de caracteríscas com diferentes tamanhos de *kernel* e profundidades. Esta característica do modelo permite a identificação de objetos em várias escalas de tamanho (LIU et al., 2016). Para classificar os possíveis objetos de uma determinada imagem, o processo de convolução percorre o mapa de características gerados para a mesma utilizando diferentes tamanhos de *kernel*, e para cada *kernel* gerados no mapa são previstas mais quatro caixas delimitadoras com deslocamento relativo a mesma região (Figura 13), além de calcular a pontuação que indica a presença para cada classe do modelo.

Figura 13: Caixas delimitadoras geradas pelas previsões dos *kernels* da rede.



Fonte: Adaptado de Liu et al. (2016).

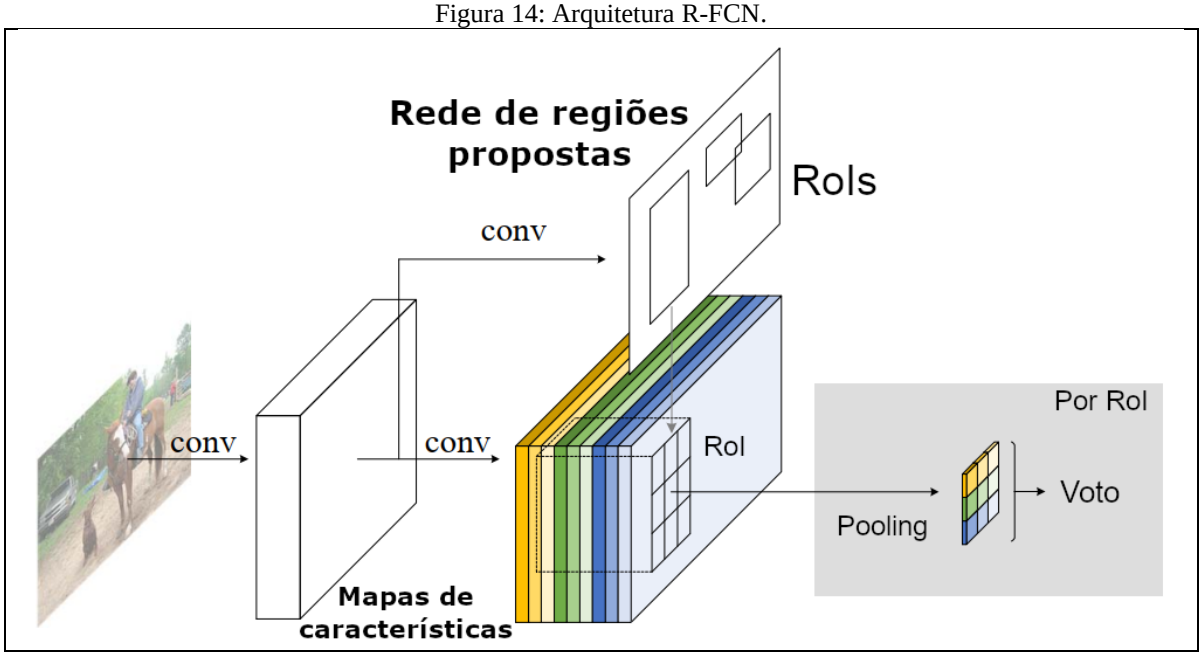
Como o resultado final de todo o processo gera um grande número de caixas delimitadoras dos possíveis objetos encontrados na imagem, bem como os respectivos valores de confiança para as predições, foi necessário implementar uma forma de delimitar o número de itens preditos. Portanto, Liu et al. (2016) adotaram o método *non-maximum suppression*[5] para definir um valor mínimo de confiança que irá determinar se uma caixa delimitadora pode ou não ser considerada como uma predição verdadeira.

2.2.7 *Region-based Fully Convolutional Networks* (R-FCN)

Dai et al. (2016), propuseram um modelo de rede para detecção de objetos semelhante ao R-CNN, ou seja, possui duas etapas, uma para localização das regiões propostas e outra para classificação destas. Contudo abordaram uma arquitetura de redes convolucional totalmente conectada, ao compartilhar os recursos gerados entre as duas etapas do modelo, conforme pode ser observado na representação da Figura 14.

[5] *Non-maximum suppression* termo em inglês para supressão não máxima.

Figura 14: Arquitetura R-FCN.



Fonte: Adaptado de Dai et al. (2016).

Para cada região de interesse gerada a partir da etapa de localização, o modelo R-FCN calcula a probabilidade de ocorrência para o número de classes definidas durante o treinamento mais o background para cada posição relativa da grade espacial k * k. O final do processo resulta em 9 pontuações de classificação cuja a classificação geral é a média destas pontuações, que por sua vez responderá se o objeto foi detectado corretamente.

2.3 SEGMENTAÇÃO SEMÂNTICA

A tarefa da segmentação semântica consiste em atribuir um rótulo de classificação para cada pixel na imagem (SHANMUGAMANI, 2018), conforme apresentado no exemplo da Figura 15. Para atribuir estes rótulos de classificação aos pixels, torna-se necessário implementar recursos precisos de identificação dos contornos dos objetos separando-os em segmentos distintos. Esta metodologia definida para construção deste tipo de recurso, faz com que a arquitetura do modelo seja mais rigorosa do que a própria detecção de objetos com caixas delimitadoras (LATEEF; RUICHEK, 2019).

Figura 15: Classificação dos pixels da imagem.

Fonte: Adaptado de DeepLab (2018).

2.3.1 DeepLab

O DeepLab é considerado até o momento como o estado da arte dos modelos de segmentação semântica. O mesmo foi desenvolvido por pesquisadores da área de visão computacional da Google, a sua distribuição como código aberto ocorreu em 2018 e atualmente encontra-se na versão v3+.

Este modelo de segmentação é composto basicamente por duas etapas. A etapa de codificação que consiste na extração de informações essenciais da imagem utilizando uma CNN pré-treinada, como por exemplo, a localização dos objetos. E a etapa de decodificação, que faz uso das informações extraídas para reconstruir a saída nas dimensões originais da imagem de entrada (LATEEF; RUICHEK, 2019).

Para contextualizar as etapas que envolvem o modelo, serão apresentados nas subseções a seguir as técnicas implementadas na construção da arquitetura do DeepLab.

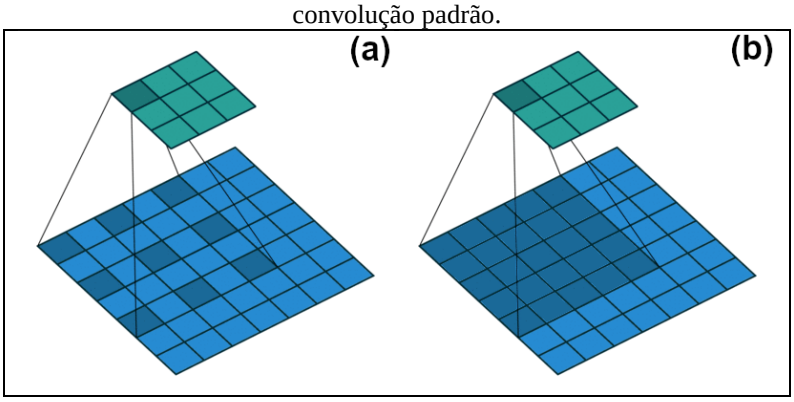
2.3.1.1 Convolução dilatada

Os primeiros modelos de redes totalmente convolucionais implementadas para a segmentação semântica demonstrou-se eficiente e proporcionou bons resultados. No entanto, o uso excessivo das operações de *pooling* em consecutivas camadas convolucionais reduziram

significativamente a resolução dos mapas de características. Esta característica limita o uso deste modelo em diferentes escalas de imagens (Chen et al., 2017).

Portanto, os idealizadores do DeepLab, adotaram a técnica de convolução dilatada[6], que altera o campo de visão do *kernel* inserindo um parâmetro extra para definir a taxa de dilatação. Conforme pode-se observar o exemplo desta operação na Figura 16, ao utilizar um *kernel* de 5x5 com taxa de dilatação 1, a operação de convolução dilatada gera um campo de visão de 9 pixels enquanto na operação convencional ocupa 25 pixels.

Figura 16: Tipos de convoluções. (a) convolução dilatada; (b) convolução padrão.

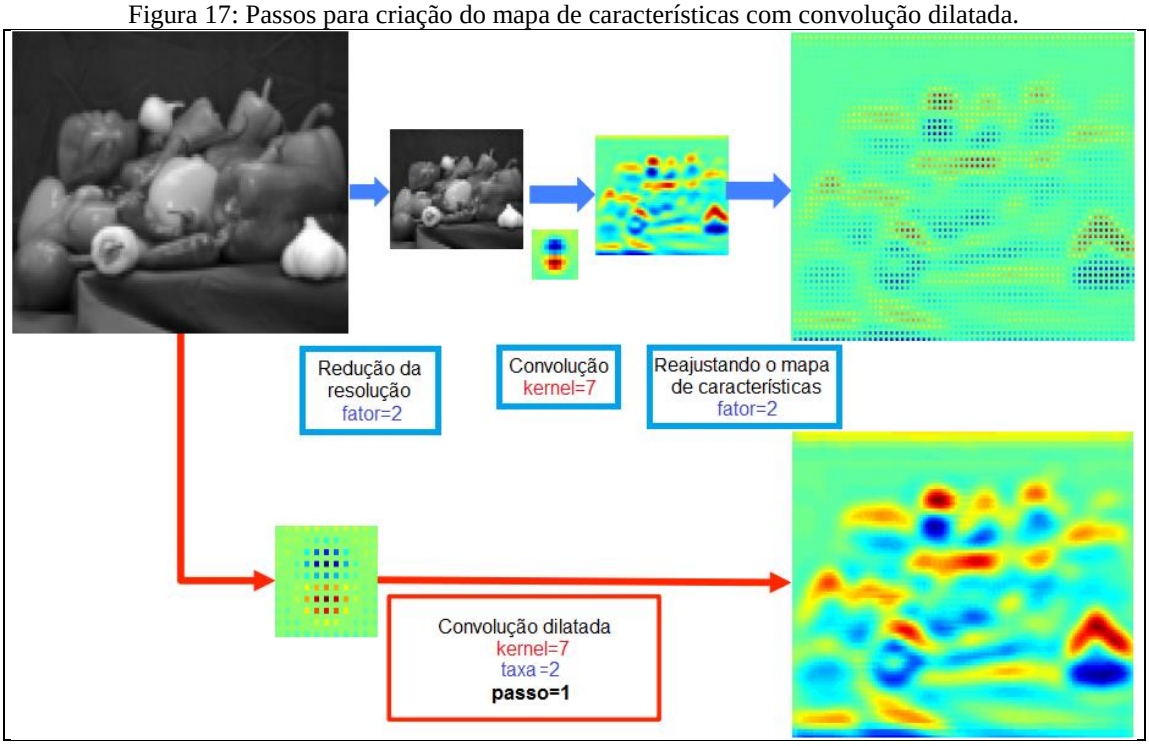


Fonte: Adaptado de DeepLab (2018).

Chen et al., (2017) relatam que a adoção desta técnica permite calcular as respostas das camadas extratoras de características de uma rede para qualquer escala de mapas geradas pelas mesmas, conforme pode ser observado através de um exemplo básico de funcionamento da mesma na Figura 17. Dado uma imagem de entrada, primeiramente aplica-se a redução de seu tamanho por um fator de 2, em seguida aplica-se uma operação de convolução padrão. O mapa de características gerado pela operação passa por um filtro com furos do mesmo tamanho da imagem de entrada para ajustar os pixels do mapa de características ao mesmo tamanho, onde os espaços vazios remanescentes são preenchidos com o valor zero. Por sua vez a convolução dilatada analisa a imagem no tamanho original gerando o mapa correspondente aos valores vazios, os resultados das operações são somados para gerar o mapa de características final.

[6] Convolução dilatada, uma tradução informal para o termo em inglês *Atrous convolution*.

Figura 17: Passos para criação do mapa de características com convolução dilatada.



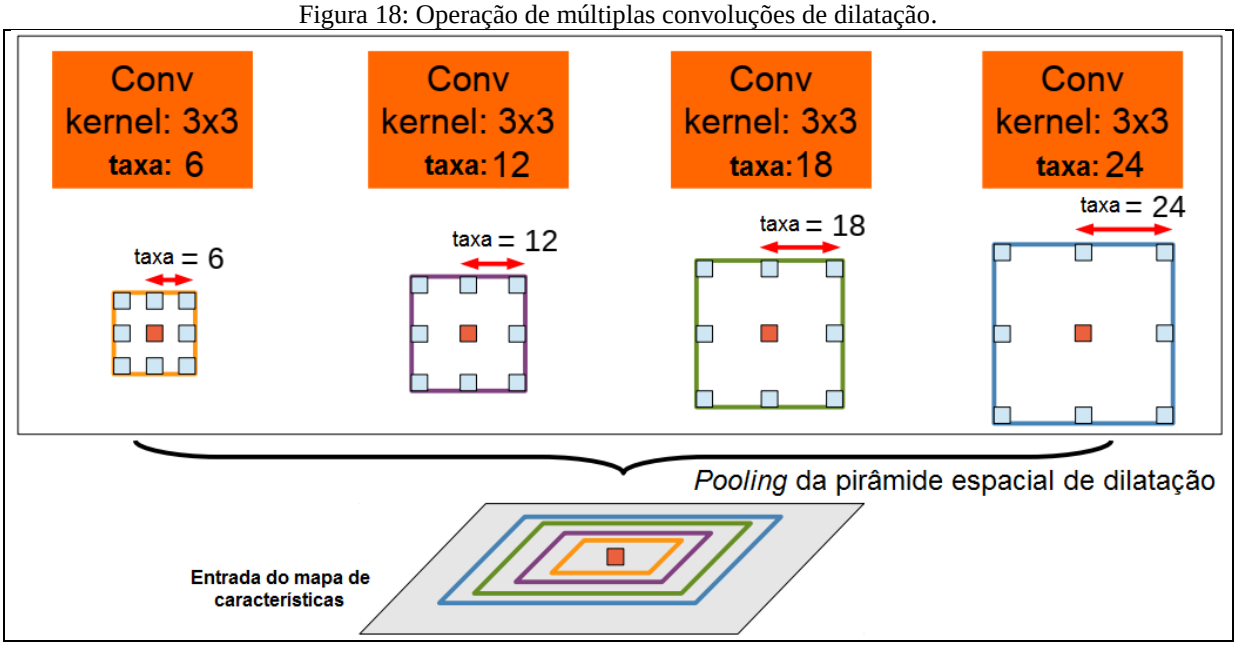
Fonte: Adaptado de Chen et al., (2017).

2.3.1.2 *Pooling* da pirâmide espacial de dilatação

Considerando que um mesmo tipo de objeto pode ser representado por diferentes escalas em uma imagem, Chen et al. (2017) introduziram ao DeepLab a técnica de *pooling* da pirâmide espacial de dilatação[7] para a etapa de codificação. A operação une paralelamente as convoluções de dilatação com diferentes taxas de dilatação na entrada do mapa de características para reconstruir as informações geradas em diferentes tamanhos nas camadas de convolução padrão (Figura 18).

[7] Pooling da pirâmide espacial de dilatação, tradução do termo em inglês Atrous Spatial Pyramid Pooling (ASPP)

Figura 18: Operação de múltiplas convoluções de dilatação.



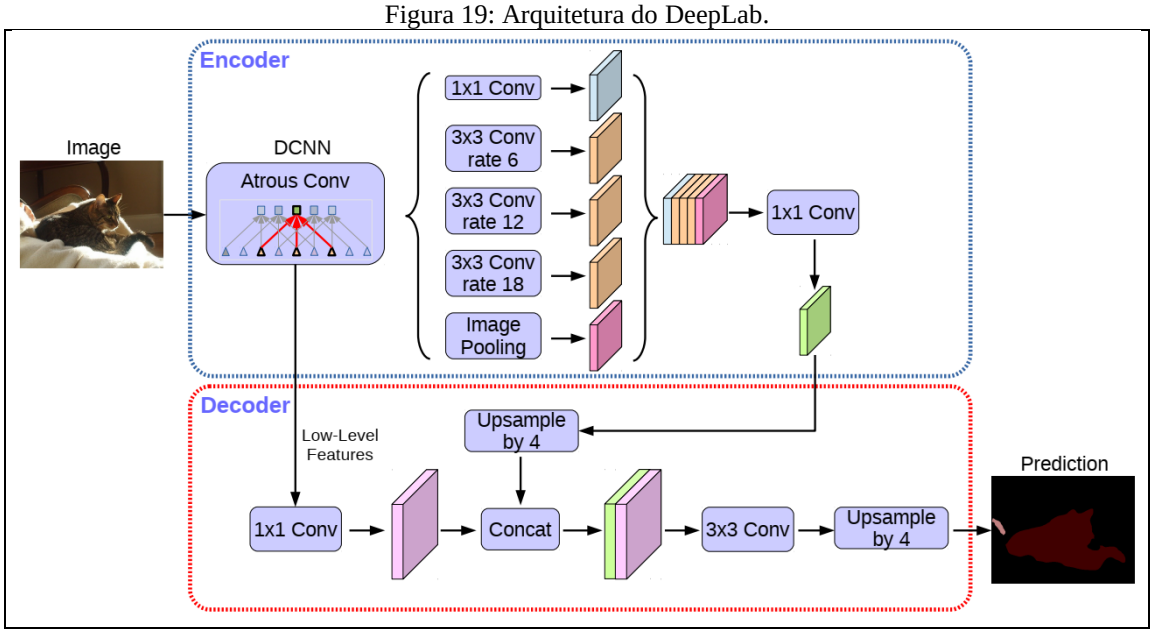
Fonte: Adaptado de Chen et al., (2017).

2.3.1.3 Decodificador

Chen et al. (2018) implementaram em seu trabalho, o recurso de decodificação para refinar o resultado da segmentação no contorno dos limites dos objetos. A arquitetura da recente versão do DeepLab, é descrita através do diagrama da Figura 19.

Como pode-se observar, o decodificador faz uso das camadas de características de baixo nível obtidas através das operações de convoluções de dilatação, que posteriormente passa por uma operação básica de convolução 1x1 para concatenar com o resultado da etapa de codificação, seguido de algumas convoluções 3x3 para refinar as características dos objetos e finalizando com o redimensionamento de fator 4 para remontar os segmentos ao tamanho original da imagem.

Figura 19: Arquitetura do DeepLab.



Fonte: Chen et al., (2018).

2.4 MATTING

Wang e Cohen (2007) descrevem que o *matting* é uma técnica de visão computacional destinada a separação precisa de um determinado objeto em primeiro plano (*foreground*) do contexto residual em segundo plano (*background*)[8]. Uma tradução literal de *matting* para português seria fosqueamento, ou seja, tornar o contexto do *foreground* fosco, que também pode ser interpretado como a produção de uma camada de transparência (*alpha*). A camada *alpha* define a opacidade dos pixels do *foreground*, através da faixa de valores entre 0 e 1 (BODA; PANDYA, 2018).

Conforme Boda e Pandya (2018), as técnicas de *matting* estão divididas em três categorias:

- Baseado em amostragem: trata-se de algoritmos como o de Chuang et al. (2001), que trabalham com a estimativa de cores entre *foreground* e *background* para calcular a cada pixel *alpha*, os modelos clássicos desta categoria trabalham com o relacionamento existente entre as amostras de pixels vizinhos e os parâmetros *alpha*. Já os métodos

[8] Os respectivos termos em inglês para primeiro plano e plano de fundo *foreground* e *background*, serão utilizados no decorrer da dissertação sempre que for necessário contextualizar a separação entre um objeto e o plano de fundo.

otimizados coletam um conjunto de pixels próximos ao *foreground* e *background* utilizando-os para adaptar o algoritmo *matting*;

- Baseado em propagação: este tipo de técnica foca na propagação da camada *alpha* através de modelos híbridos que fazem uso dos algoritmos baseados em amostragem. Como é o caso do algoritmo *Knn* criado por Chen, Li e Tang (2013), que aborda esta metodologia ao determinar que os pixels vizinhos ao pixel avaliado incorporem os valores *alpha* semelhantes ao mesmo. Neste mesmo contexto estão os algoritmos *Closed form* de Levin, Lischinski e Weiss (2007) e *Large Kernel Matting* (*Lkm*) de He, Sun e Tang (2010), que operam em uma área em torno do pixel avaliado para determinar o fluxo de dados presente no local e propagar os valores adequadamente, algo semelhante acontece no algoritmo *Information Flow Matting* (*Ifm*) de Aksoy, Aydin e Pollefeys, 2017), que controla o fluxo de dados das regiões de *foreground* e *background* para a região de intersecção através das definições de afinidade entre os pixels; e
- Baseado em aprendizado: as técnicas baseadas em aprendizado local aprendem sobre a distribuição dos pixels *alpha* vizinhos do que está sendo estimado para gerar o valor do mesmo. Nos casos de aprendizado global como o modelo implementado por Zheng e Kambhamettu (2009), o algoritmo aprende sobre algum pixel previamente rotulado que esteja mais próximo ao pixel avaliado para adequar-se ao melhor *matting* baseado em *trimap*.

2.4.1 *Trimap*

O *trimap* é um mapa de segmentos que divide a imagem em três regiões definitivas, *foreground*, *background* e região desconhecida ou a área de intersecção entre o objeto e o restante da cena. Esta abordagem é utilizada na maioria dos algoritmos *matting*, e permite delimitar a região de interesse que deve ser processada pelo algoritmo (WANG; COHEN, 2007). A Figura 20 demonstra os passos da operação de *matting* utilizando o *trimap*.

Figura 20: Exemplo da operação de *matting*.

Fonte: Adaptado de Wang e Cohen (2007).

Comumente o *trimap* é gerado manualmente pelo usuário e a definição da área de intersecção deve ser precisa, cobrindo apenas os pixels que realmente interessam para o contexto do objeto. Portanto, quanto mais fino o *trimap*, menor será o número de pixels que deverá ser estimado.

Contudo, como este trabalho almeja a construção de um processo totalmente automatizado, a criação do *trimap* será guiado com base na linha gerada pela etapa de segmentação, ou seja, a linha do segmento substituirá a descrição manual da área de intersecção. A metodologia adotada para a criação do *trimap* está descrita na seção 4.3.

2.5 MÉTRICAS DE AVALIAÇÃO

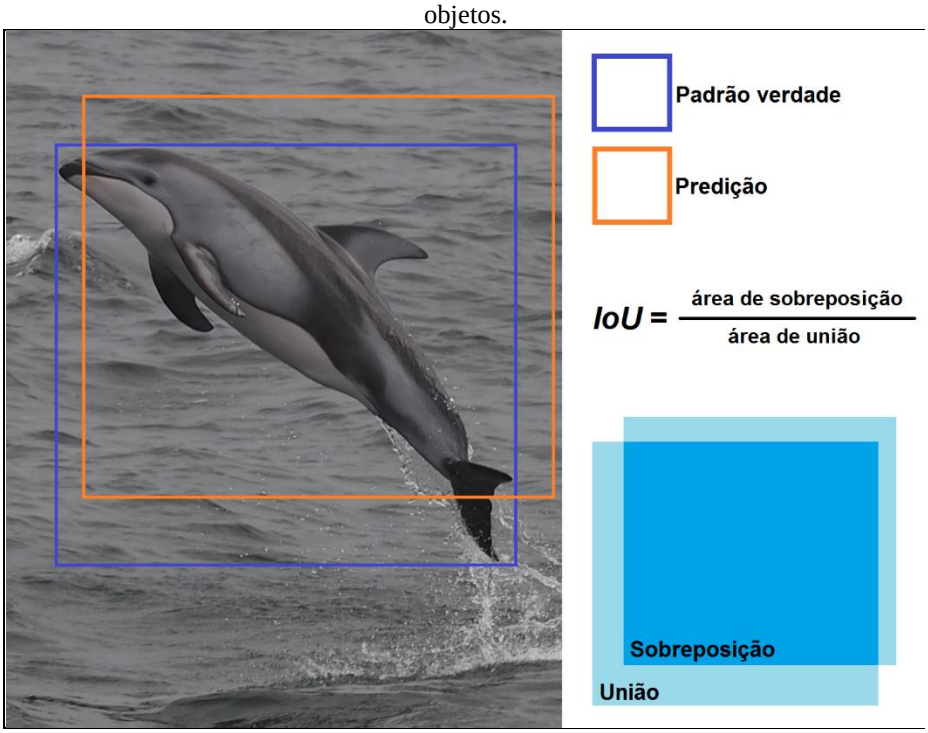
Conforme Shanmugamani (2018), as métricas de avaliação são de extrema importância para as tarefas de aprendizado de máquina, pois permitem entender o comportamento dos modelos criados ao avaliar o quão preciso são ao executar uma determinada tarefa.

Nas tarefas de detecção de objetos a métrica comumente utilizada é a Precisão Média também conhecida pelo termo em inglês *Average Precision* (AP). No entanto, para avaliar se o recurso de detecção de objetos localizou corretamente um determinado objeto em uma imagem, utiliza-se a unidade de medida conhecida como Intersecção Sobre a União ou *Intersection Over Union* (IoU) (SHANMUGAMANI, 2018).

2.5.1 *Intersection Over Union* (IoU)

O IoU avalia o percentual de sobreposição de duas caixas delimitadoras, sendo uma destas caixas o padrão verdade criado a mão por uma pessoa e a outra resultante da detecção de objetos, conforme pode ser observado na Figura 21 (EVERINGHAM et al., 2010).

Figura 21: Avaliação da sobreposição das caixas delimitadoras para a detecção de objetos.



Fonte: Compilação do autor.

Portanto, o valor resultante do IoU é dado pela Equação (1). Onde B_p é a caixa prevista, B_{gt} é a caixa do padrão verdade, $B_p \cap B_{gt}$ denota a interseção e $B_p \cup B_{gt}$ a união.

$$a_o = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (1)$$

Através do valor resultante é possível declarar se uma determinada detecção do modelo é verdadeira ou falsa. Para isto basta definir um limiar de corte, como por exemplo, o valor 0,5 definido pelo PASCAL VOC (EVERINGHAM et al., 2010), ou a escala de valores entre 0,5 à 0,95 determinado pelo conjunto de métricas de avaliação do COCO (2015).

Durante a avaliação dos resultados para este trabalho o IoU também foi adotado como métrica de avaliação da etapa de segmentação. Contudo, conforme será apresentado na seção 5.1.2 do capítulo 5 , os valores obtidos representam a média ponderada para IoU (mIoU).

2.5.2 *Average Precision* (AP)

Conforme Everingham et al. (2010), o AP resume a forma da curva de precisão e revocação[9] a partir dos resultados de classificação para um determinado método, ou seja, o AP é basicamente a média da precisão sobre todos os valores de revocação entre 0 e 1. Para melhor entendimento de como o valor de avaliação é obtido, será necessário explicar os conceitos que envolvem a obtenção dos resultados da curva de precisão e revocação.

2.5.2.1 Precisão e revocação

A precisão mede a porcentagem de previsões verdadeiras positivas que foram encontradas por um determinado modelo, cujo valor é obtido através da Equação (2). Já a revocação quantifica a capacidade de um modelo prever todas as informações relevantes e seu valor é dado pela Equação (3). Sendo *tp* os verdadeiros positivos ou previsões corretas, valor definido pelo limiar de corte da avaliação do IoU (e.g. $\geq 0{,}5$). O *fp* é o equivalente aos falsos positivos ou detecções erradas IoU (e.g. $< 0{,}5$) e o *fn* são as definições do padrão verdade que não foram previstas pelo modelo (EVERINGHAM et al., 2010).

$$Precisão = \frac{tp}{tp + fp} \tag{2}$$

$$Revocação = \frac{tp}{tp + fn} \tag{3}$$

2.5.2.2 AP - PASCAL VOC

O PASCAL VOC foi uma competição voltada a avaliação de desempenho de modelos de detecção de objetos e segmentação que ocorreu entre os anos de 2005 a 2012, onde incorporou a métrica AP a partir de 2007 como o método de avaliação padrão para as detecções de objetos (EVERINGHAM et al., 2010).

---

[9] Revocação é uma tradução não literal para o termo em inglês *Recall*.

O cálculo de AP definido pela Equação (4), infere a interpolação dos valores de precisão sobre os 11 pontos de revocação entre 0 e 1. Onde r é a precisão medida no ponto de revocação r'.

$$AP = \frac{1}{11} \sum_{r \in \{0,0.1,\ldots,1\}} Pinterp^{(r)} \tag{4}$$

onde

$$Pinterp^{(r)} = \max_{r':r' \geq r} p(r')$$

A métrica AP do PASCAL VOC é calculada para todas as classes presentes em um modelo preditivo, no entanto um valor global também é calculado ao final do processo de avaliação. Trata-se da média ponderada sobre os produtos de AP para todas as classes do modelo (mAP).

2.5.2.3 AP - COCO

O cálculo de AP para a métrica de avaliação COCO (2015) compartilha de alguns dos conceitos da avaliação do PASCAL VOC, as diferenças ficam para o número de interpolações que sai de 11 para 101 (e.g. {0,0.01,...1}), além do valor de mAP que é gerado a partir de 10 valores de IoU entre 0,5 à 0,95.

2.5.3 *F-score*

O *F-score* ou medida *F*, mede a precisão de um determinado teste baseando-se na média harmônica da curva de precisão e revocação (MARTIN; FOWLKES; MALIK, 2004). E seu valor é dado pela Equação (5).

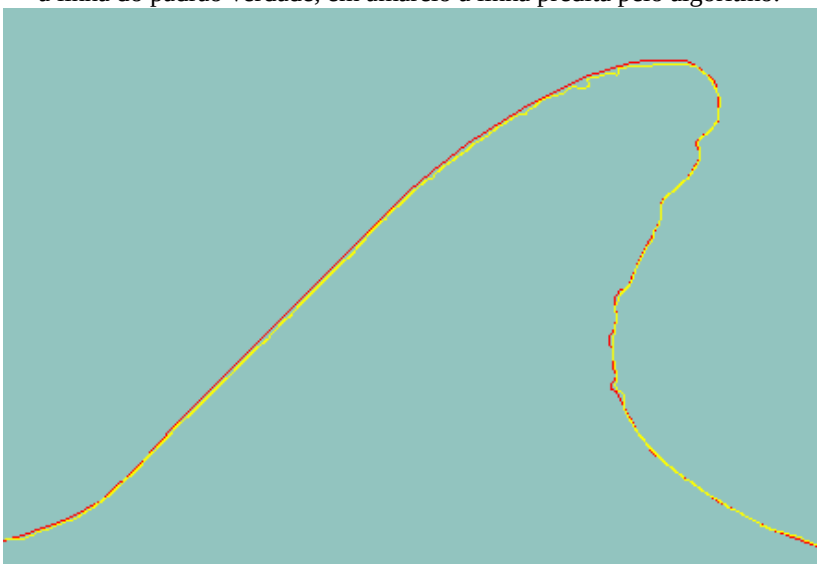
$$F = 2 * \frac{precisão * revocação}{precisão + revocação} \tag{5}$$

Este método é comumente utilizado na avaliação de técnicas de detecção de contorno, como por exemplo, o algoritmo desenvolvido por Canny (1986).

No entanto, ao avaliar os pixels de um contorno gerado pela tarefa e o contorno criado como padrão verdade por um humano, a probabilidade de se obter a correspondência exata destes pixels é baixa, principalmente quando comparado a uma única amostra de padrão verdade (Figura 22). Portanto, este tipo de abordagem pode declarar que os resultados obtidos pela tarefa de detecção são imprecisos, mesmo que este tenha gerado contornos utilizáveis (ARBELAEZ et al., 2010).

Prevendo este tipo de limitação Martin, Fowlkes e Malik (2004), criaram um algoritmo de avaliação de detecção de contorno mais flexível que a medida *F-score* padrão, ou seja, esta avaliação tende a ser mais tolerante no que diz respeito a pequenos deslocamentos da linha detectada.

Figura 22: Correspondências dos pixels da linha de contorno. Em vermelho a linha do padrão verdade, em amarelo a linha predita pelo algoritmo.



Fonte: Compilação do autor.

Portanto, neste trabalho optou-se por utilizar a métrica criada por Martin, Fowlkes e Malik (2004) durante a avaliação da etapa de extração da linha de contorno, pois os resultados obtidos não se enquadram ao modelo de avaliação restrito do *F-score* padrão.

3 TRABALHOS RELACIONADOS

Buscando apresentar o estado da arte dos trabalhos voltados a identificação individual de mamíferos marinhos da ordem dos cetáceos, foi realizada uma pesquisa seguindo alguns conceitos da revisão sistemática.

A pesquisa do tema foi realizada em fevereiro de 2018 e focou na consulta dos repositórios de artigos, IEEE, Science Direct, ACM, Springer Link, Scopus e Google Scholar.

Como critério de inclusão definiu-se que os artigos deveriam ser todos em inglês e não seria adotado o critério temporal relacionado ao período de publicação, devido a necessidade de contextualizar a evolução das soluções adotadas para a resolução do problema proposto até o presente momento. Por outro lado, o critério de exclusão que foi estabelecido ignora qualquer trabalho cuja as características adotadas para a identificação individual não sejam provenientes das nadadeiras dorsais dos indivíduos.

Outro objetivo da pesquisa realizada é responder os seguintes questionamentos:

- Q1: Quais trabalhos fazem uso de técnicas de visão computacional para extrair as características necessárias para identificação individual a partir da linha de contorno da dorsal dos indivíduos?
- Q2: Dos trabalhos que satisfazem a Q1, quais realizam o processo de extração das características do contorno da dorsal sem a interação humana, ou seja, automatiza o processo?
- Q3: entre os trabalhos avaliados, quais aplicam testes de validação do desempenho para etapa de extração das características do contorno da dorsal?

A *string* de busca foi construída pensando encontrar artigos voltados principalmente na identificação individual baseada em fotografias tiradas das nadadeiras dorsais dos indivíduos.

String de busca:

- *("photo-id" OR "photo identification" OR "individual identification" OR "animal identification" OR "animal biometric") AND ("fin" OR "dorsal") AND ("dolphin" OR "cetacea" OR "delphinidae").*

O Quadro 1 apresenta dos resultados obtidos com as buscas efetuadas nos repositórios.

Quadro 1. Lista de títulos dos trabalhos relacionados.

Repositório	**Quantidade de artigos retornados na consulta**	**Quantidade de artigos encontrados relacionados ao tema do trabalho**
IEE	4	1
Science Direct	247	0
Springer Link	160	3
ACM	0	0
Scopus	72	0
Google Scholar	4550	4

Considerando os termos escolhidos para definir os critérios da *string* de busca observa-se que, a pesquisa realizada no Google Scholar encontrou dois artigos que remetem ao repositório IEEE, vide Quadro 2. Contudo, as buscas nestes repositórios com a mesma *string* não retornaram os artigos em questão.

A busca realizada na Scopus retornou como resultado o trabalho de Genov et al. (2018). No entanto, este foi desconsiderado pois o método de identificação abordado no trabalho faz uso das caraterísticas faciais dos indivíduos para identificação, fugindo do escopo definido para este trabalho.

É importante ressaltar um detalhe sobre os trabalhos 7 e 8 listados no Quadro 2 como trabalhos relacionados. Apesar de tratarem da identificação individual de animais que não pertencem a ordem dos cetáceos, ao comparar as nadadeiras dorsais de um tubarão branco e um golfinho é possível observar a semelhança morfológica entre as diferentes espécies, viabilizando a implementação da metodologia apresentada nesses trabalhos para os dois tipos de animais.

Quadro 2. Lista de títulos dos trabalhos relacionados.

Nº	Artigo	Autor/Ano	Repositório
1	"Finscan", a Computer System for Photographic Identification of Marine Animals	Hillman et al., (2002)	IEEE [10]
2	Unsupervised Thresholding for Automatic Extraction of Dolphin Dorsal Fin Outlines from Digital Photographs in DARWIN	Hale (2008)	Google Scholar
3	Computer-assisted Recognition Of Dolphin Individuals Using Dorsal Fin Pigmentations	Gilman et al. (2016)	IEEE
4	Photo-id of blue whale by means of the dorsal fin using clustering algorithms and color local complexity estimation for mobile devices	Carvajal-Gámez et al. (2017)	Springer Link
5	Integral Curvature Representation and Matching Algorithms for Identification of Dolphins and Whales	Weideman et al. (2017)	IEEE [1]
6	Wild Cetacea Identification using Image Metadata	Pollicelli, Coscarella e Delrieux (2017)	Google Scholar
7	Automated Visual Fin Identification of Individual Great White Sharks	Hughes e Burghardt (2016)	Springer Link
8	Semi-automated software for dorsal fin photographic identification of marine species: application to Carcharodon carcharias	Andreotti et al. (2017)	Springer Link

A seguir será explorado com mais detalhes os trabalhos citados no Quadro 2.

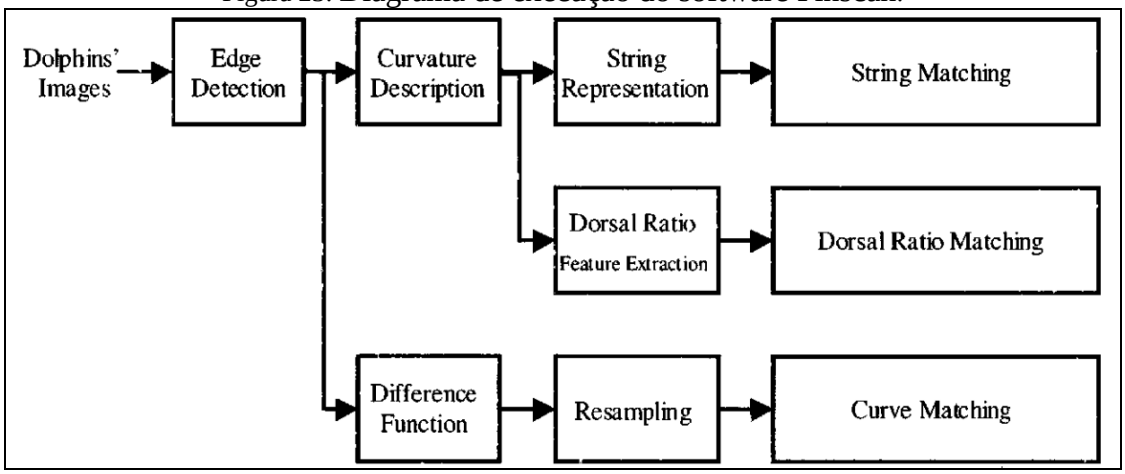
3.1 FINSCAN, UM SISTEMA DE IDENTIFICAÇÃO FOTOGRÁFICA PARA ANIMAIS MARINHOS

O Finscan foi um software desenvolvido para auxiliar os pesquisadores no processo de identificação de animais marinhos como, golfinhos, baleias e tubarões (HILLMAN et al., 2002).

O software foi alterado ao longo de anos, almejando sempre a evolução do processo de identificação individual (ARAABI et al., 2000). A Figura 23, apresenta o diagrama das etapas de execução do software.

[10] Artigos mantidos pela IEEE, porém foram encontrados durante a pesquisa no Google Scholar.

Figura 23: Diagrama de execução do software Finscan.



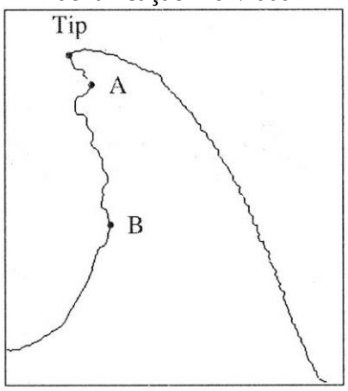
Fonte: Fonte: Araabi et al. (2000).

Os dois primeiros módulos apresentados no diagrama da Figura 23 foram desenvolvidos e descritos no trabalho de Kreho et al. (1997). No primeiro módulo foi implementada a técnica de detecção de contornos *Laplacian of Gaussian* (LoG), já o segundo módulo focou na implementação do descritor da curvatura.

O trabalho de Kreho et al. (1997), também abordou dois métodos de comparação as linhas de contorno.

O método *Dorsal Ratio Matching*, conhecido como um método manual de identificação *Dorsal Ratio* (DR), determina que a comparação é realizada a partir da distância entre os dois entalhes mais significantes da dorsal dividido pela distância do menor entalhe encontrado no topo da mesma, conforme representado na Figura 24. No entanto, este método não era eficiente quando confrontado a dorsais de indivíduos que apresentem mais que dois entalhes significativos.

Figura 24: DR, método manual de identificação individual



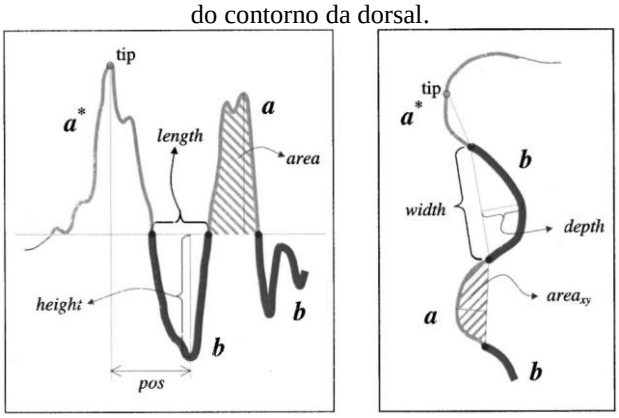
Fonte: Araabi et al. (2000).

Portanto, para suprir esta deficiência os autores implementaram o *Curve Matching* que apresentou resultados melhores que o DR. Este método utiliza uma função de diferença para avaliar a semelhança entre as curvas dos entalhes da dorsal ao comparar dois indivíduos.

Contudo, percebeu-se que o modelo de comparação de curvas em algumas ocasiões contribuiu mais ao avaliar entalhes menos significativos do que os mais significativos. Por este motivo Hillman et al. (2002), resolveram adotar o método de comparação baseado em uma representação de cadeia de caracteres "*string*".

O método em questão considera que os entalhes no contorno da dorsal dos indivíduos podem ser representados por funções de curvatura, onde atributos de medição como largura, comprimento, altura, profundidade, área, posição podem ser representados por notações primitivas como "a" e "b" (Figura 25).

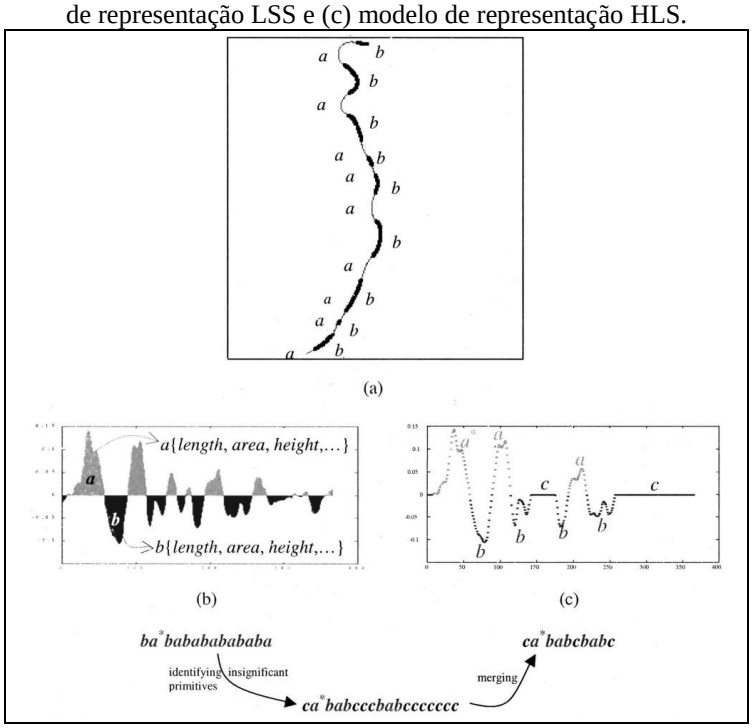
Figura 25: Notações primitivas e atributos de medidas do contorno da dorsal.



Fonte: Araabi et al. (2000).

Dois modelos de notação foram implementados, o L*ow-level String Representation* (LLS) e *High-level String Representation* (HLS) baseado no LLS. O LLS caracteriza-se como uma representação tolerante a flutuações ou ruídos, já o HLS mescla as características insignificantes, considerando apenas as representações mais significativas geradas pelo LSS. A Figura 26 denota a forma de representação de cada modelo.

Figura 26: (a) notações primitivas do contorno da dorsal, (b) modelo de representação LSS e (c) modelo de representação HLS.



Fonte: Araabi et al. (2000).

Para a etapa de comparação e identificação de indivíduos foi criado um método de medida das distâncias sintáticas/semânticas para as representações primitivas LSS e HLS. O método é modelado no trabalho pela Equação (6).

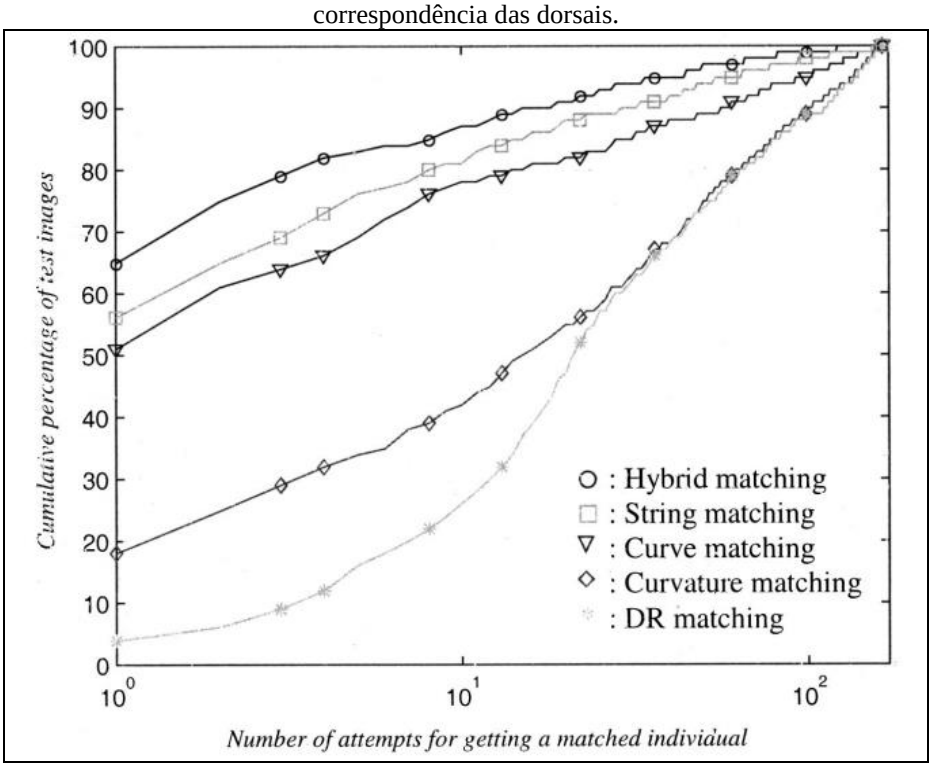
$$D_{att}(s_1, s_2) = \sum_i \Delta_i\,(att)(d_i\,/\,d)w_i \tag{6}$$

Onde i é o index dos valores primitivos, $(d_i\,/\,d)$ é o comprimento normalizado de cada primitivo, w_i é o peso de dependência da sintaxe com base na matriz dos pesos de substituição e Δ_i (att) é o peso de dependência semântica que reflete na diferença de valores entre os atributos de duas *strings*.

Os testes realizados por Araabi et al. (2000), envolveram um conjunto de 624 imagens para uma população de 164 golfinhos encontrados no Golfo do México, destes oito indivíduos possuíam apenas uma imagem e por este motivo foram descartados da base de dados de testes. Para o restante dos indivíduos uma imagem de cada foi selecionada para realizar a comparação com as demais imagens inseridas na base de testes da aplicação.

A avaliação dos resultados abrange os testes considerando os métodos DR e a comparação da curvatura proposto no trabalho de Kreho et al. (1997), bem como o método de comparação de *string* e uma versão hibrida que junta a comparação de curva com a *string*. Os resultados obtidos podem ser observados na Figura 27, o eixo vertical do gráfico define a porcentagem de imagens classificadas corretamente como top-1 do *ranking*, já o eixo horizontal define o número de indivíduos examinados antes de encontrar a primeira correspondência válida durante a comparação de dorsais.

Figura 27: Resultados obtidos por Araabi et al. (2000), ao avaliar os algoritmos de correspondência das dorsais.



Fonte: Araabi et al. (2000).

3.2 *THRESHOLD* NÃO SUPERVISIONADO PARA EXTRAÇÃO AUTOMÁRICA DA LINHA DE CONTORNO DA DORSAL DE GOLFINGOS ATRAVÉS DE FOTOGRAFIAS DIGITAIS NO SOFTWARE DARWIN

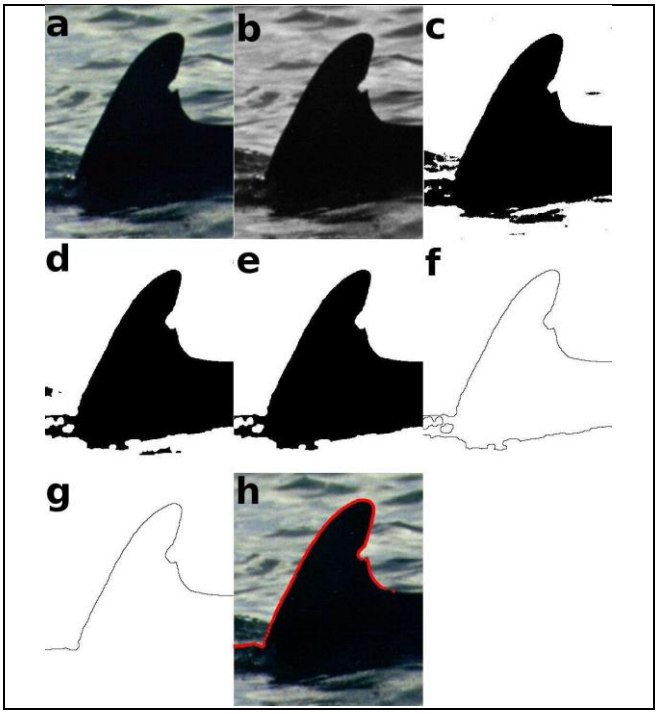
O trabalho de Hale (2008), consiste em criar um método semiautomático de extração do contorno da dorsal dos golfinhos, utilizando a técnica de segmentação conhecida como *threshold* não supervisionado.

O método proposto pelo autor adota duas abordagens distintas, com intuito de reduzir ao máximo o trabalho manual de seleção do contorno da dorsal dos golfinhos. A Figura 28 apresenta os passos para extração da linha de contorno.

A primeira abordagem trabalha com a imagem colorida, sem executar qualquer tipo de alteração antes do processamento. Esta abordagem é executada em três etapas:

- Binarização: com a imagem transformada em escala de cinza, o processo efetua a análise do histograma para determinar o valor do *threshold*. Ao encontrar o primeiro vale do histograma seleciona-se o valor da constante de intensidade da binarização;
- Processo morfológico: aplica-se técnicas de erosão e dilatação para remoção dos ruídos; e
- Seleção do contorno da dorsal: seleciona o maior elemento após o processo morfológico, uma cópia deste elemento é criada e aplica-se o procedimento de erosão uma única vez, em seguida os dois elementos são comparados utilizando o operador lógico (XOR) que resulta na linha de contorno da dorsal.

Figura 28: Passos para a etapa de geração do contorno da dorsal.



Fonte: Hale (2008).

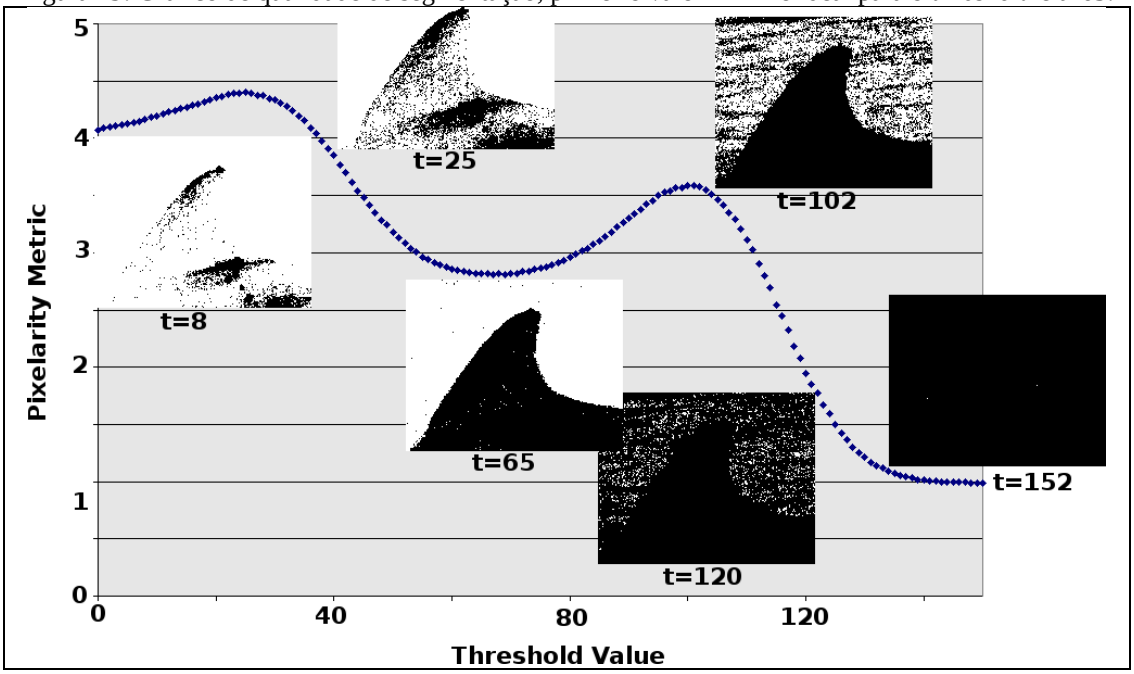
O autor descreve que com a primeira abordagem, de 302 imagens processadas 96 não produziram linhas de contorno passível de serem utilizadas na etapa de identificação de indivíduo. Esta falha ocorreu devido a incidência da reflexão de luz nas dorsais.

Portanto, para suprir esta deficiência o autor criou uma segunda abordagem, que consiste em utilizar a cor ciano do sistema de cores CMYK para encontrar o valor do *threshold* para a etapa de binarização da imagem.

Quase todo o processo da segunda abordagem tem como base as etapas apresentadas na primeira abordagem, com exceção da etapa de binarização. O autor adotou uma métrica de avaliação da capacidade de construção de áreas sólidas através do método de segmentação.

Está métrica avalia as conexões de cada pixel com os seus vizinhos determinando um peso. Uma média é gerada a partir dos pesos, o que permite avaliar a qualidade da segmentação. Este processo é aplicado para os valores de *threshold* que variam de 0 à 160, onde o valor do primeiro local mínimo é selecionado como o ideal para a etapa de binarização (Figura 29).

Figura 29: Gráfico de qualidade de segmentação, primeiro valor mínimo ideal para o *threshold* é t=65.



Fonte: Hale (2008).

Os resultados apresentados pelos autores descrevem que de 302 imagens utilizadas para teste 35.1% geraram linhas úteis para o processo de identificação sem a necessidade de modificação, 33.11% das imagens precisaram de algum tipo de modificação posterior para gerar linhas úteis. Das imagens restantes 31,79%, ou seja, 96 imagens resultaram em linhas inconsistentes para o processo de identificação.

Já para a segunda abordagem os autores utilizaram 94 imagens diferentes do primeiro conjunto de imagens aplicadas na primeira abordagem. Onde 48 imagens (51%) produziram linhas úteis para o processo de identificação, sem aplicar qualquer tipo de modificação na imagem.

Apesar do bom resultado apresentado no trabalho, é importante destacar que o processo de extração do contorno da dorsal do golfinho não é totalmente automatizado.

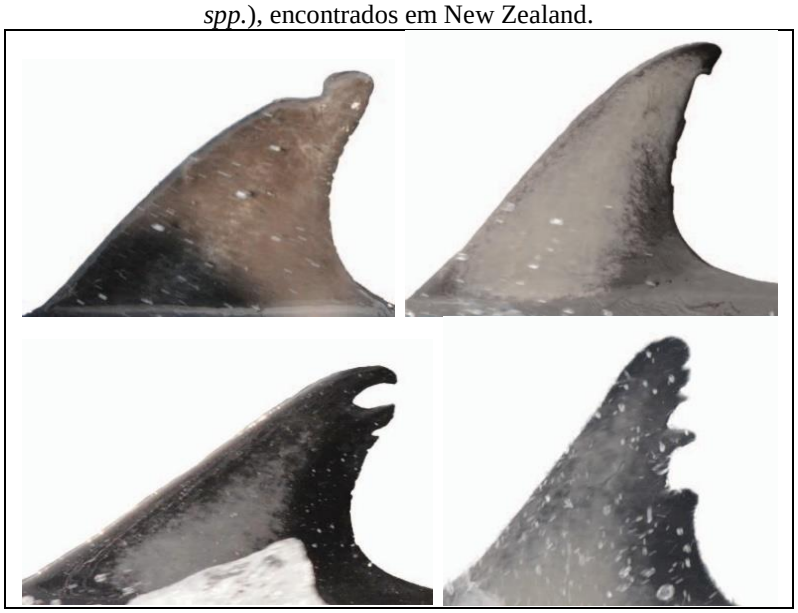
O autor deixa claro no texto que, antes de executar o processo, o usuário é obrigado a informar o ponto de início e término da dorsal. Esta ação permite delimitar a área de busca, reduzindo a chance de erros e melhorando a qualidade das informações do histograma da imagem.

3.3 RECONHECIMENTO INDIVIDUAL DE GOLFINHOS UTILIZANDO A PIGMENTAÇÃO DA DORSAL

Este trabalho adotou uma abordagem oposta aos demais trabalhos encontrados na literatura, ou seja, fez uso da pigmentação presente na dorsal do golfinho como atributo do processo de identificação individual (Figura 30).

Conforme Gilman et al. (2016), a abordagem escolhida para a identificação individual se justifica, pois, a pigmentação encontrada nos golfinhos não apresenta bordas ou pontos pontiagudos, isso permite trabalhar com métodos de quantificação de pigmentação.

Figura 30: Pigmentação na dorsal dos golfinhos comuns (*Delphinus spp.*), encontrados em New Zealand.

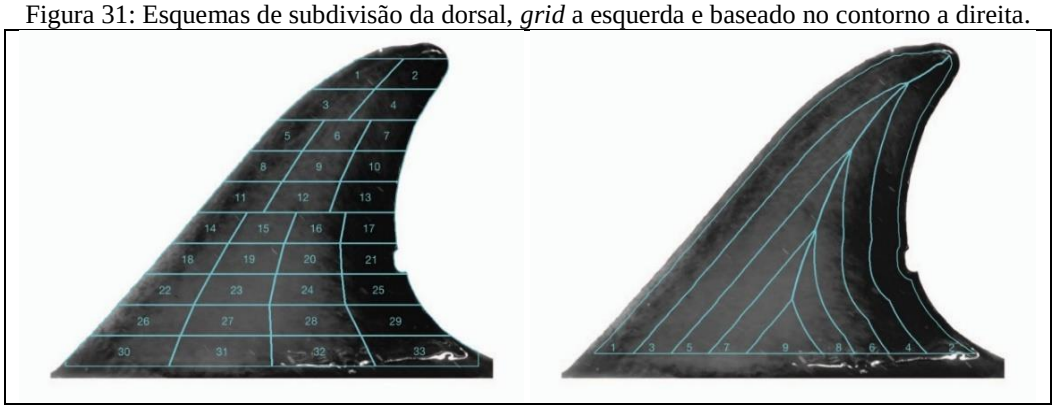


Fonte: Gilman et al. (2016).

Considerando que os valores das características para cada indivíduo a ser identificado viriam da pigmentação da dorsal, os autores decidiram aplicar um esquema de subdivisão da dorsal em áreas menores, permitindo empregar um processo estatístico robusto para obtenção dos valores utilizados na etapa de identificação.

A subdivisão é aplicada a duas abordagens, sendo a primeira com base na distância mais próxima da borda, já a segunda consiste em segmentos de grade ao longo do eixo perpendicular a base da dorsal, conforme apresentado pela Figura 31.

Figura 31: Esquemas de subdivisão da dorsal, *grid* a esquerda e baseado no contorno a direita.



Fonte: Gilman et al. (2016).

Os passos seguintes do processo de extração de características envolvem, a normalização de valores dos pixels da imagem convertida para escala de cinza, subtraindo a média e dividindo pelo desvio padrão de todos os pixels. Bem como, aplicando o cálculo estatístico da média, mediana e intervalo interquartil nas 42 subdivisões, seguido de vários métodos de desvio padrão.

A etapa de identificação do indivíduo foi construída pensando na classificação das imagens dos indivíduos e divide-se em duas etapas:

- *Ranking* das características: criação do subconjunto de características que representarão o indivíduo durante a identificação. A correlação dos dados do conjunto é avaliada através do método de correlação t-score.
- Treinamento: a etapa de classificação fez uso da técnica estatística de combinação linear *Linear Discriminant Analysis* (LDA), sendo o *ranking* da classificação determinado pelo método de validação cruzada *Leaveone-out Cross Validation* (LOOCV).

A avaliação dos resultados da classificação mostrou que ao utilizar a divisão da dorsal baseado em *grid*, proporcionou um acerto de 71.2% ao identificar corretamente o indivíduo (top-1) e 83.7% de acerto para o nível de predição de características similares com até cinco indivíduos diferentes (top-5). Para o caso de subdivisão baseado no contorno da dorsal, a classificação resultou em um acerto de 53.5% (top-1) e 77.3% (top-5).

No entanto o melhor resultado foi juntando os dois tipos de divisão que ficou em 75.5% (top-1) e 86.3% (top-5). Comprovando a eficiência do método proposto para o caso de estudo ao adotar as duas metodologias.

3.4 FOTO IDENTIFICAÇÃO DE BALEIA AZUL PARA DISPOSITIVOS MÓVEIS ATRAVÉS DA NADADEIRA DORSAL USANDO ALGORITMOS DE CLUSTERING E ESTIMATIVA DE COMPLEXIDADE LOCAL DAS CORES

Almejando a criação de um sistema portátil e prático para uso em áreas remotas de monitoramento e pesquisa de baleias azuis (*Balaenoptera musculus*), Carvajal-Gámez et al. (2017) desenvolveram um aplicativo para dispositivos móveis com sistema operacional Android que auxilia na identificação individual desses animais, através das nadadeiras dorsais.

Este aplicativo buscou atender a questão relacionada ao poder de processamento limitado dos dispositivos móveis, bem como o consumo excessivo de recursos, como por exemplo, bateria e espaço de armazenamento de dados. A técnica de segmentação de imagem proposta para o trabalho, precisa aplicar filtros de aprimoramento das linhas de contorno com o intuito de reduzir o tamanho das imagens e melhorar o tempo de processamento, porém sem perder o detalhamento do objeto. Além de implementar um método de redução das paletas de cores presentes na imagem, diminuindo a quantidade de pixels de difícil classificação.

Para alcançar o objetivo do trabalho, os autores dividiram a técnica de segmentação em cinco estágios.

Estágio 1 - banco de imagens: as imagens utilizadas no trabalho foram obtidas por dispositivos móveis de diferentes marcas e modelos, com câmeras fotográficas de resoluções que variam entre 5 e 13 MP. Todas as imagens foram obtidas e processadas no padrão de cores RGB.

Estágio 2 - pré-processamento: faz uso do filtro de passa-banda *Discrete Wavelet Transform* (DWT) para descrever a textura do corpo da baleia, bem como aplica o algoritmo de análise de sinais conhecido como *Circular Haar Wavelet* (CHW), compactando a imagem sem perder a informação relacionada as linhas de contorno do animal. O pré-processamento é executado separadamente para cada camada do padrão RGB.

Estágio 3 - redução da paleta de cores: buscando o desempenho da aplicação em relação ao tempo de processamento e uso de espaço para armazenamento da informação, este estágio introduz um método de remoção de corres redundantes existentes em cada canal RGB da imagem, aplicando a técnica de quantificação de pixel descrito no trabalho de Carvajal-Gamez, Gallegos-Funes e Rosales-Silva (2013).

Estágio 4 - segmentação adaptativa: visando melhorar a nitidez e o contraste da cena ao destacar o objeto procurado do fundo da imagem, foi introduzido um método de segmentação sobre cada canal de cor R, G e B, utilizando um filtro de histograma dinâmico combinado à técnicas de análise de *cluster*.

Estágio 5 - minimizando o número de pixels classificados incorretamente: o último estágio é responsável pela segmentação final da imagem, primeiro convertendo os canais RGB resultantes do estágio 4 em valores de tons de cinza, finalizando com a binarização.

O método proposto pelo trabalho, contempla a segmentação e classificação da imagem, separando o indivíduo avistado do restante da cena. Contudo não aborda a etapa de identificação do indivíduo, nem a de extração das características individuais do animal como, a linha de contorno da dorsal ou pigmentação do corpo.

Para avaliar a performance da técnica de segmentação desenvolvida, os autores realizaram testes de validação tendo como parâmetro de medida imagens segmentadas manualmente.

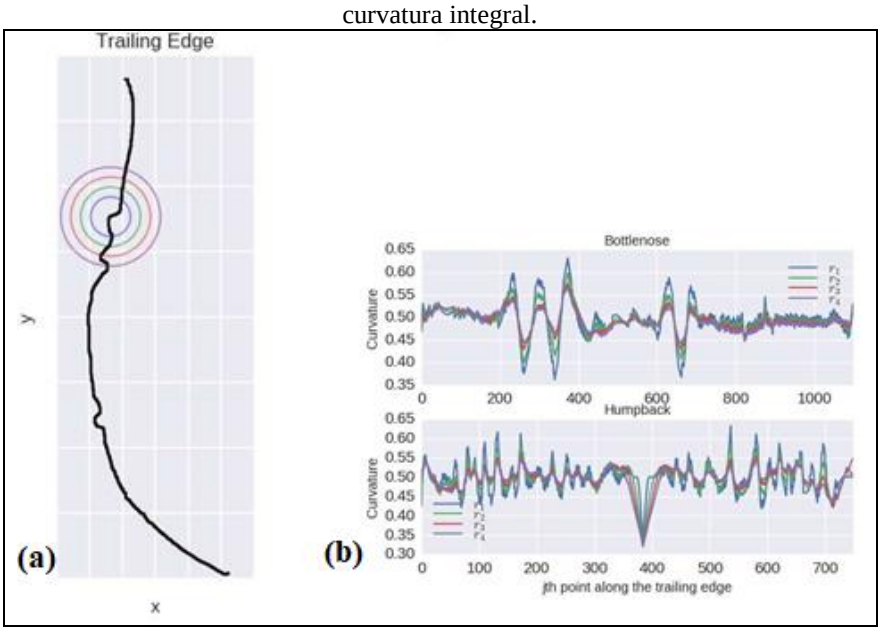
O catálogo de imagens para teste envolve 771 imagens distribuídas em três categorias, imagens de animais com dorsal triangular, deitada e curvada. Os resultados apresentam uma acurácia no método de segmentação implementado que variam de 98.97% à 99.04% para a dorsal triangular, 95.30% à 95.49% deitada e 97.56% à 98.05% curvada.

3.5 REPRESENTAÇÃO DA CURVATURA INTEGRAL E ALGORITMOS DE CLASSIFICAÇÃO PARA IDENTIFICAÇÃO DE GOLFINHOS E BALEIAS

Weideman et al. (2017), propuseram em seu trabalho a medida da curvatura integral para a representação e extração da linha de contorno das dorsais de golfinhos e das caudas de baleias, em conjunto com dois algoritmos para classificação e identificação individual dos animais.

O método da medida de curvatura integral proposto pelos autores, busca tornar a representação da linha de contorno das dorsais e caudas dos indivíduos mais robusta. Ou seja, pretende-se definir uma métrica confiável de conversão das linhas de contorno para uma curvatura linear no eixo horizontal, independente do ponto de vista ou pose em que o indivíduo se encontra na imagem (Figura 32).

Figura 32: Conversão de curvatura da linha de contorno. (a) exemplo de linha do contorno de uma dorsal de golfinho, (b) conversão da linha de contorno em curvatura integral.



Fonte: Adaptado de Weideman et al. (2017).

A etapa que antecede a conversão das linhas de contorno para o método proposto no trabalho, visa a extração destas aplicando uma técnica de segmentação baseada em Rede Neural Convolucional proposto por Long, Shelhamer e Darrell (2015). A imagens foram previamente recortadas pelos pesquisadores antes da execução deste procedimento, reduzindo o esforço de análise do algoritmo de segmentação.

Na etapa de identificação dos indivíduos os autores aplicaram duas técnicas para comparação das linhas de contorno dos indivíduos, produzindo um ranking de indivíduos com características similares ao indivíduo consultado, deixando a cargo do pesquisador a seleção da correspondência exata entre os indivíduos avaliados.

A primeira técnica é a *Dynamic Time-Warping* (DTW), que permite comparar as representações temporais de duas curvaturas, calculando o custo de alinhamento entre elas.

Já a segunda técnica aplicada fez uso de um método de classificação chamado de *Local Naive Bayes Nearest Neighbor* (LNBNN) (LOWE; MCCANN, 2012), que originalmente foi criado para classificar imagens. Contudo, neste trabalho o foco foi a classificação do conjunto de dados transformados em descritores de características e pontos chave as linhas de contorno.

Os testes da metodologia proposta foram realizados com um conjunto de 10713 imagens para 401 indivíduos distintos de golfinhos nariz de garrafa (*Tursiops truncatus*) e 7173 imagens de 3572 indivíduos de baleia-jubarte (*Megaptera novaeangliae*).

Deste conjunto de imagens os autores selecionaram aleatoriamente para cada golfinho arquivos de 10 encontros, nos casos onde o indivíduo apresentou um número inferior a 10 de encontros selecionou-se arquivos para n-1 encontro. Para montar a base de dados das baleias, as amostras selecionadas representavam em sua maioria uma imagem por indivíduo devido ao baixo número de encontros.

Os resultados apresentados no trabalho indicam que, a identificação individual dos golfinhos obteve melhor resultado com a técnica DTW que corresponde à acurácia de 74% no top-1 do ranking e 69% com LNBNN. No entanto o oposto ocorreu para os testes dos dados das baleias onde a técnica LNBNN apresentou 89% de acurácia no top-1 e 86% para DTW.

3.6 IDENTIFICAÇÃO DE CETACEOS UTILIZANDO METADADO

Pollicelli, Coscarella e Delrieux (2017), buscaram validar a hipótese de identificação individual dos golfinhos da espécie *Cephalorhynchus commersoni*, através dos metadados levantados a partir da avaliação de imagens tiradas de 223 indivíduos ao longo de sete anos.

A abordagem não fez uso de técnicas de visão computacional para extração das marcas características encontradas nas dorsais destes indivíduos, ou seja, todas as informações necessárias para o estudo foram levantadas através da avaliação minuciosa dos pesquisadores para cada foto tirada.

Conforme descrito pelos autores, os seguintes atributos foram selecionados nos metadados coletados:

- Lado: posição que o animal foi fotografado “direito” ou “esquerdo”;
- Qualidade: índice de 0 a 3 para efeitos de brilho, contraste e condição visual da dorsal;
- Distinção: também um índice de 0 a 3 para o quão distinguível estava as marcas das dorsais;
- Cicatrizes: quantidade visíveis na dorsal;

- Coloração: quantidade de pontos com coloração anormais presentes na dorsal;
- Zonas: divisão da dorsal em áreas para definir onde as marcas estavam presentes;
- Entalhes: quantidade de entalhes presentes no contorno da dorsal;
- Tamanhos das marcas: descrição da quantidade de marcas com as seguintes definições de tamanhos, grande, comprida, estendida, média, pequena/minúscula; e
- Formato de marcas: descrição da quantidade e formato das marcas com as seguintes definições, pouco, leve, imperceptível, triangular arredondada e saliente.

Os métodos de classificação selecionados pelos autores para a etapa de identificação do indivíduo foram:

- Redes neurais: *Multilayer Perceptron*;
- Classificador bayesiano: *NaïveBayes*;
- Árvore de decisão: *J48*; e
- Algoritmo do K-vizinho mais próximo: KStar.

No entanto, uma avaliação preliminar foi efetuada utilizando *Info Gain Attribute Eval*, *Gain Ratio Attribute Eval* e *Chi Squared Attribute Eval* em conjunto com o método de busca *Ranker*, para encontrar a relevância dos atributos escolhidos. Alguns foram descartados devido à baixa relevância para o processo de classificação.

O passo seguinte do trabalho foi a construção e validação dos modelos. Do conjunto de dados contendo 869 instancias de 223 indivíduos, foram criados dois subconjuntos que apresentavam o número de capturas entre 5 – 12 e maior ou igual a cinco, resultado respectivamente em 373 instancias de 54 indivíduos e 515 instâncias de 62 indivíduos.

Os subconjuntos foram testados estatisticamente com o classificador *ZeroR*, o qual demostrou através do resultado de 2,4862%, que os dados classificaram corretamente, melhor até que ao puro acaso que corresponde ao valor de 1,8%.

A execução dos testes foi aplicada aos dois subconjuntos removendo respectivamente 10% e 3% dos dados, simulando novos encontros de indivíduos. Conforme apresentado na Tabela 1 os

resultados variaram de 72,72% à 90% para o conjunto de 3% de dados. Já para o caso contendo 10% das instâncias a variação dos resultados ficou entre 56,86% e 72,97%.

Tabela 1. Resultados obtidos para a etapa de identificação de indivíduos com base nos modelos escolhidos para a classificação de metadados.

Dados de validação	Base de dados	*Naive Bayes*	*KStar*	*J48*	*Multilayer Perceptron*
10%	**5 a 12**	72.97%	75.67%	70.27%	70.27%
	≥ 5	68.62%	62.74%	64.70%	56.86%
3%	**5 a 12**	81.81%	81.81%	90%	72.72%
	≥ 5	87.5%	87.5%	81.25%	81.25%

Fonte: Adaptado de Pollicelli, Coscarella e Delrieux (2017).

3.7 IDENTIFICAÇÃO INDIVIDUAL AUTOMATIZADA DE TUBARÕES BRANCOS

Considerado como o primeiro trabalho voltado ao tema de identificação de indivíduos através da nadadeira dorsal, que funciona de modo totalmente automatizados. Hughes e Burghardt (2016), desenvolveram uma proposta que contempla a execução das duas etapas principais de um software de identificação individual.

Na etapa de extração das características necessárias para a identificação, foi adotada uma abordagem de segmentação e detecção de contorno baseado em mapas ultra métricos de contorno e descritores de atributos de componentes presentes na imagem. Em seguida foi desenvolvido um método de codificação biométrica baseado na suavização de objetos, adequando a região de contorno ao formato do contorno da dorsal dos tubarões brancos.

Conforme descrito pelos autores, o modelo de detecção de contorno e detecção de objetos candidatos à dorsal divide-se em três estágios:

Estágio 1, segmentação: utilizando como base o trabalho de Arbeláez et al. (2014), que trata de uma abordagem para segmentação hierárquica, que para o trabalho em questão proporciona um conjunto de 200 regiões segmentadas, o qual posteriormente é classificado novamente para retiradas de regiões pequenas de mais para serem consideradas como uma dorsal, sobrando apenas 12 regiões segmentadas por imagem analisada.

Estágio 2, geração de candidatos a dorsal: como a imagem contempla pelo menos a dorsal de um indivíduo, também considerando que a dorsal se trata de um contorno aberto que facilmente mescla ao restante do corpo formando um único objeto, aplicou-se o método de detecção de cantos proposto por Zhang et al. (2009) na definição dos pontos de início e término da linha de contorno da dorsal.

Estágio 3, *ranking* dos candidatos a dorsal: o último estágio é responsável pelo treinamento do classificador *Random Forest Regressor* (BREIMAN, 2001), que prevê a qualidade de hipótese de possíveis dorsais computadas pelo *framework* de detecção e avaliação de contorno BSDS (MARTIN; FOWLKES; MALIK, 2004). Para o treinamento do classificador foram utilizadas 240 imagens de alta visibilidade e 120 imagens de baixa visibilidade, cujas as linhas de contorno dos indivíduos foram delimitadas manualmente para criação dos descritores de características.

Para validar a abordagem da primeira etapa foi adotado o teste proposto por Hariharan et al. (2014), que visa medir a performance da segmentação e detecção dos objetos candidatos a dorsal. Os resultados obtidos podem ser observados na Tabela 2 e Tabela 3.

Tabela 2. Resultados intermediários dos testes de desempenho para a detecção de objetos.

	t=0.7	t=0.85	t=0.9	AP^{vol}
Segmentation	1.0	0.99	0.99	0.99
Candidate gen. (H)	0.99	0.98	0.98	0.97
Candidate gen. (L)	1.0	0.99	0.92	0.96

Fonte: Adaptado de Pollicelli, Coscarella e Delrieux (2017).

Tabela 3. Resultados dos testes de desempenho para a detecção da dorsal.

Feature type	t=0.7	t=0.85	t=0.9	AP^{vol}
High Visibility (H)				
OpponentSIFT	0.99	0.85	0.73	-
Normal	0.98	0.85	0.7	-
Combined	0.98	0.95	0.86	0.92
Lower Visibility (L)				
Combined	1.0	0.93	0.62	0.89

Fonte: Adaptado de Pollicelli, Coscarella e Delrieux (2017).

A codificação biométrica da linha de contorno da dorsal pretende aumentar a precisão da identificação individual, aplicando um refinamento no contorno com um método de máscara de

opacidade proposto por Zheng e Kambhamettu (2009), seguido pela definição dos pontos chaves que descrevem a linha reaplicando o método descrito no Estágio 3, finalizando com a implementação de descritores de características semi-locais e globais.

Para a etapa de identificação do indivíduo, os autores abordaram a técnica de classificação *Local Naive Bayes Nearest Neighbor* (LNBNN) (LOWE; MCCANN, 2012). Esta técnica permite interpretar os descritores de características de um indivíduo e comparar com os descritores de indivíduos que já possuem identificação e encontram-se armazenados em uma base de dados.

Como resultado deste processo espera-se o retorno de um ranking de dorsais similares ao indivíduo analisado, possibilitando ao pesquisador a identificação correta do novo indivíduo, que terá os dados armazenados no banco de dados da aplicação.

Os testes foram aplicados a um conjunto de 2456 imagens de 85 indivíduos, uma imagem de cada indivíduo foi separada para montar o conjunto de testes sobrando 2371 imagens que foram utilizadas para montar a base de dados de identificação.

Os resultados apresentados demonstraram que em 82% dos casos os indivíduos foram identificados corretamente, sendo que em 91% das vezes a identificação correta encontrava-se entre os dez primeiros do ranking. Apenas 9% das instancias não foram classificadas corretamente.

3.8 SOFTWARE SEMI-AUTOMATIZADO PARA IDENTIFICAÇÃO DE INDIVÍDUOS DA ESPÉCIE CARCHARODON CARCHARIAS ATRAVÉS DE FOTOGRAFIAS DA NADADEIRA DORSAL

Com intuito de construir um software específico para identificação individual de tubarão branco Andreotti et al. (2017) desenvolveram uma proposta semelhante ao software DARWIN.

Esta semelhança se caracteriza pelo fato de que o software proposto foi desenvolvido pensando em uma aplicação desktop, que permite construir uma base de dados para comparação dos indivíduos e exige que o pesquisador informe os pontos de início e termino do contorno da dorsal antes de passar para a etapa de identificação individual.

No entanto as técnicas aplicadas nas duas etapas do processo de identificação do indivíduo diferem do software DARWIN.

Na etapa de extração das características do contorno da dorsal, os autores aplicaram a técnica de detecção de contornos Sobel apenas na área delimitada pelo usuário.

De posse da linha de contorno o software automaticamente executa a etapa de identificação individual, aplicando a técnica *Dynamic Time-Warping* (DTW) no processo de comparação da nova linha de contorno com as informações existentes no banco de dados. O processo de comparação permite criar um *ranking* de probabilidade das melhores combinações encontradas.

Os autores descrevem que a base de dados disponível para testes consiste em 744 imagens de 426 indivíduos identificados manualmente. Deste conjunto de indivíduos, 50 foram selecionados aleatoriamente para montar a base de testes do software.

Os resultados apresentados no trabalho demonstram que das 50 imagens analisadas pelo software, 40 delas foram comparadas corretamente ou seja 80%. Destas 62% encontraram o indivíduo correto nas duas primeiras posições do ranking. Contudo, não foi possível encontrar indivíduos correspondentes para 7 imagens e em um único caso a imagem não pode ser classificada.

3.9 ANÁLISE COMPARATIVA

Como complemento da seção de trabalhos relacionados o Quadro 3, apresenta uma análise comparativa resumida das principais técnicas utilizadas nos mesmos.

Quadro 3. Comparação das técnicas utilizadas para extração de características da dorsal e identificação do indivíduo, nos trabalhos relacionados.

Nº	**Artigo**	**Extração das características da dorsal**	**Classificação e identificação do indivíduo**
1	"Finscan", a Computer System for Photographic Identification of Marine Animals	*Laplacian of Gaussian* (LoG)	*Curve matching* *String matching*
2	Unsupervised Thresholding for Automatic Extraction of Dolphin Dorsal Fin Outlines from Digital Photographs in DARWIN	*Threshold* não supervisionado	Não abordado neste trabalho
3	Computer-assisted Recognition Of Dolphin Individuals Using Dorsal Fin Pigmentations	Média dos valores normalizados com base na pigmentação da dorsal	Classificação com *Linear Discriminant Analysis* (LDA) e identificação individual com *Leaveone-Out Cross Validation* (LOOCV)
4	Photo-id of blue whale by means of the dorsal fin using clustering algorithms and color local complexity estimation for mobile devices	Segmentação com filtro de passa-banda *Discrete Wavelet Transform* (DWT), análise de cluster com *Fuzzy C-means* e *K-means* nos canais RGB e finalizando com binarização da imagem em escala de cinza	Não abordado neste trabalho
5	Integral Curvature Representation and Matching Algorithms for Identification of Dolphins and Whales	Segmentação com rede neural convolucional	*Dynamic Time-Warping* (DTW) *Local naive Bayes nearest neighbor* (LNBNN)
6	Wild Cetacea Identification using Image Metadata	Metadado das características da dorsal extraídas manualmente	Redes neurais: *Multilayer Perceptron*; Classificador bayesiano: *NaïveBayes*; Árvore de decisão: *J48*; Algoritmo do K-vizinho mais próximo: *KStar*.
7	Automated Visual Fin Identification of Individual Great White Sharks	Segmentação hierárquica e identificação de objetos com *Random Forest Regressor*a	*Local naive Bayes nearest neighbor* (LNBNN)
8	Semi-automated software for dorsal fin photographic identification of marine species: application to Carcharodon carcharias	Detecção de contornos com Sobel	*Dynamic Time-Warping* (DTW)

3.10 CONSIDERAÇÕES

É possível avaliar no Quadro 3, que os trabalhos listados empregaram diferentes técnicas para a tarefa de extração de características das nadadeiras dorsais. Técnicas que vão desde a abordagem clássica como a detecção de contornos com Sobel de Andreotti et al. (2017), até o desenvolvimento de técnicas hibridas como Hughes e Burghardt (2016).

Outro aspecto que se pode destacar, entre os oito trabalhos encontrados durante o levantamento do estado da arte, apenas dois não adotam a linha de contorno da dorsal como caraterística relevante para identificação individual.

Portanto isso faz refletir que em 75% dos trabalhos, a caraterística mais relevante para identificação do indivíduo é a linha de contorno da dorsal e consequentemente valida as afirmações levantadas pela maioria dos biólogos de que se trata da principal característica a ser utilizada no processo de identificação individual não invasivo, justificando o uso desta no escopo do trabalho.

Também é válido ressaltar que a escolha do trabalho de Hughes e Burghardt (2016) como referência para o desenvolvimento deste justifica-se, pois foi o único a apresentar uma solução totalmente automatizada para as duas etapas de um software de identificação individual, bem como apresentou testes consistentes que demonstraram a qualidade da solução proposta para a tarefa de localização da dorsal na imagem e extração da linha de contorno para a execução da etapa de comparação de indivíduos.

O capítulo a seguir apresentará um panorama detalhado das escolhas tomadas para o desenvolvimento da solução proposta para este trabalho, bem como trará as justificativas para cada técnica ou algoritmo escolhido para cada tarefa da etapa de extração das características do contorno da dorsal.

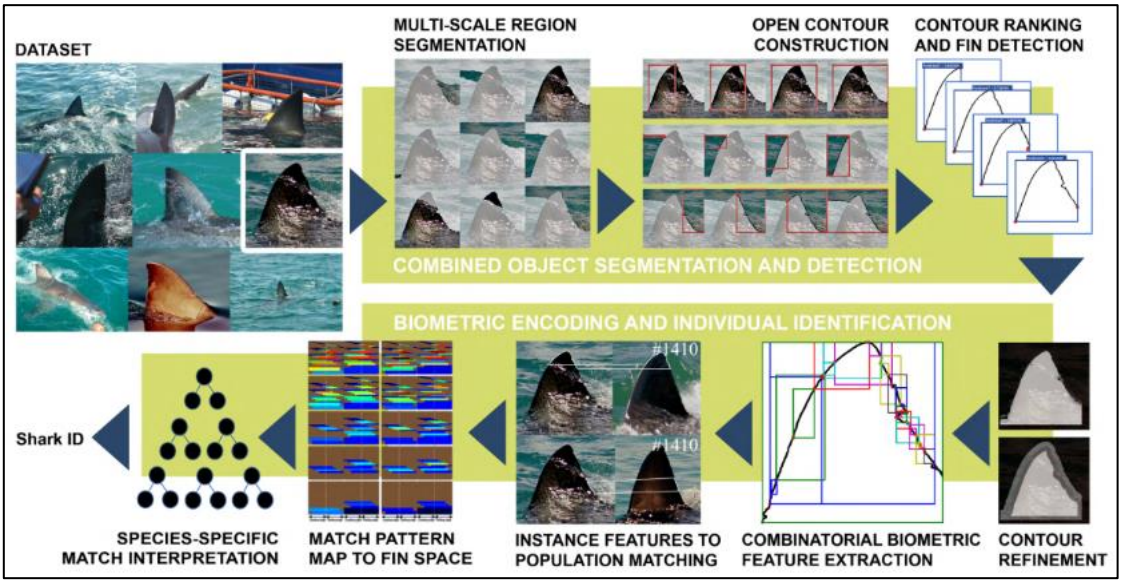
4 DESENVOLVIMENTO

Este capítulo contextualiza todas as etapas que envolvem a criação de uma ferramenta automatizada para localização e extração da linha de contorno das dorsais de cetáceos em imagens. O desenvolvimento da ferramenta foi significantemente influenciado pelo trabalho desenvolvido por Hughes e Burghardt (2016), pois foram os únicos que elaboraram um mecanismo totalmente automatizado para resolução de um problema semelhante ao que está sendo abordado neste trabalho.

Contudo, se compararmos o trabalho Hughes e Burghardt (2016) com este, pode-se notar uma visão diferenciada na criação das etapas que envolvem o processo de desenvolvimento da ferramenta. A primeira diferença trata da limitação do escopo para este trabalho, ou seja, a etapa de identificação do indivíduo não será implementada.

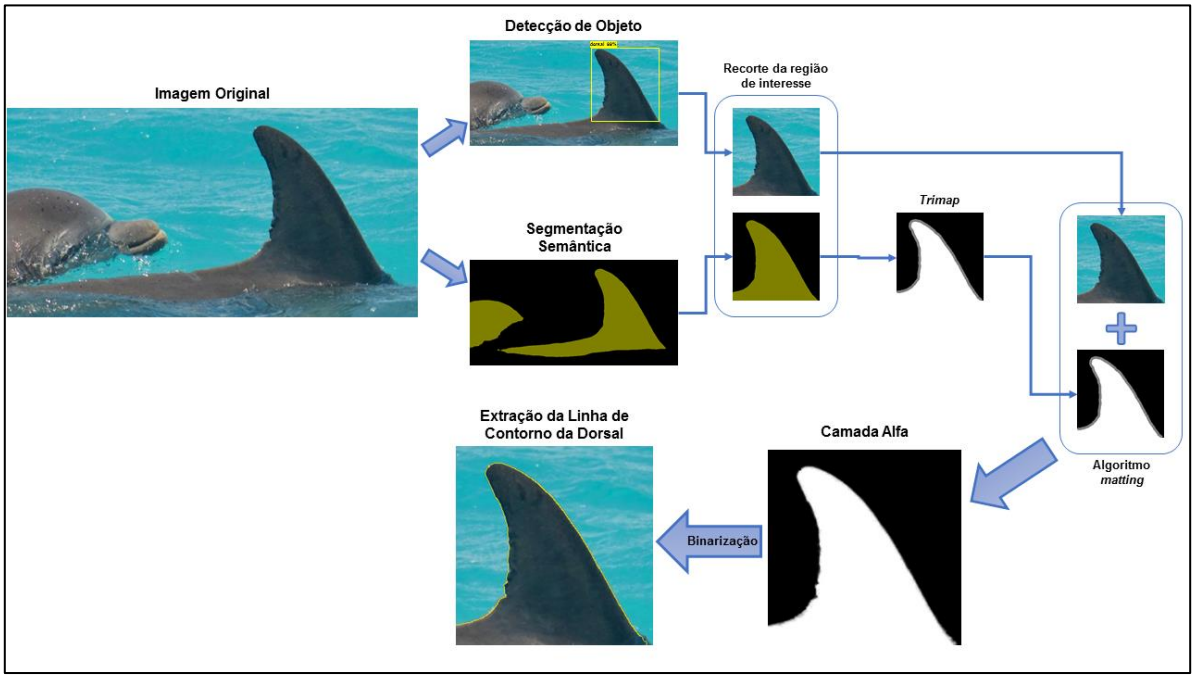
A segunda apoia-se na metodologia adotada para resolver o problema proposto. Enquanto Hughes e Burghardt (2016) optam por um processo que exige a segmentação hierárquica da imagem, passando pelo refinamento de limites dos segmentos para posterior classificação dos candidatos a dorsal utilizando *Random forests* e finalizando com o tratamento de contorno da dorsal para extração da linha e inferência do algoritmo de identificação individual (Figura 33). Este trabalho fez uso do método de detecção de objetos para localizar e delimitar as dorsais dos indivíduos nas imagens, bem como implantou-se a técnica de segmentação semântica que separa o objeto de interesse do restante da cena, finalizando com o refinamento e extração da linha de contorno da dorsal (Figura 34).

Figura 33: Modelo criado por Hughes e Burghardt (2016) para automatizar o processo de identificação individual de tubarão branco.



Fonte: Hughes e Burghardt (2016).

Figura 34: Diagrama do processo de detecção e extração da linha de contorno da dorsal para a ferramenta desenvolvida neste trabalho.



Fonte: Compilação do autor.

4.1 DETEÇÃO DE OBJETOS

Esta etapa fez uso do *framework* de código aberto TensorFlow Object Detection API criado por Huang et al. (2016), que permite localizar e identificar múltiplos objetos em uma única imagem. Trata-se de um *framework* construído na plataforma de aprendizado de máquina disponibilizado pela Google, que facilita o treinamento de redes neurais para identificação de objetos visando a construção de ferramentas cujo escopo sejam aplicações voltadas a computação visual.

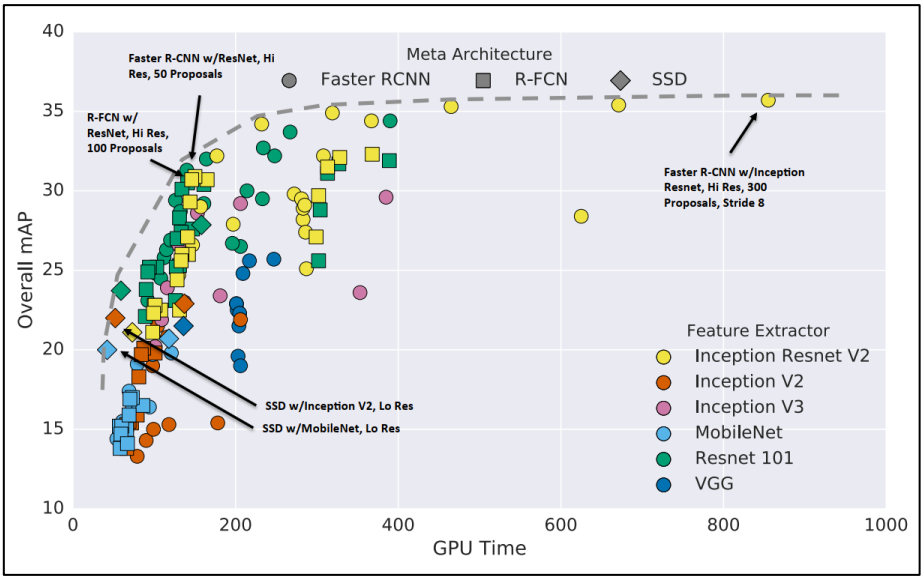
A API disponibiliza três meta-arquiteturas de redes neurais e seis possíveis extratores de características. Algumas combinações destes recursos permitiram que os autores criassem um total de 15 modelos de redes neurais, evidenciando a diversidade e flexibilidade da ferramenta proposta (HUANG et al., 2016). Tanto a diversidade de configurações disponíveis para a API, bem como os resultados significativos apresentados pelos autores, levou a integração da mesma no processo de desenvolvimento deste trabalho. O Quadro 4 apresenta as combinações de configurações criadas pelos autores.

Quadro 4. Combinação de meta-arquiteturas e extratores de características utilizados no trabalho de Huang et al. (2016).

Meta-arquitetura	**Extratores de características utilizados**
Faster R-CNN (REN et al., 2015)	VGG-16 (SIMONYAN; ZISSERMAN, 2014) Resnet-101 (HE et al., 2015) Inception v2 (LOFFE; SZEGEDY, 2015) Inception v3 (SZEGEDY et al., 2015) Inception Resnet (SZEGEDY et al., 2016) MobileNet (HOWARD et al., 2017)
R-FCN (DAI et al., 2016)	Resnet-101 (HE et al., 2015) Inception v2 (LOFFE; SZEGEDY, 2015) Inception Resnet (SZEGEDY et al., 2016) MobileNet (HOWARD et al., 2017)
SSD (LIU et al., 2016)	VGG-16 (SIMONYAN; ZISSERMAN, 2014) Resnet-101 (HE et al., 2015) Inception v2 (LOFFE; SZEGEDY, 2015) Inception Resnet (SZEGEDY et al., 2016) MobileNet (HOWARD et al., 2017)

As combinações entre meta-arquitetura e extratores de características, permitiu os autores da API realizarem testes distintos com diversas configurações de tamanho de entrada, número de passos, etc (HUANG et al., 2016). A Figura 35 apresenta os resultados obtidos onde é possível observar em alguns casos o equilíbrio entre precisão e tempo de processamento, bem como uma representação significativa na curva de aprendizado dos modelos.

Figura 35: Precisão x tempo, cada forma geométrica representa a meta-arquitetura e as cores os extratores de características.



Fonte: Huang et al. (2016).

Bem como descrito por Huang et al. (2016) em seu trabalho, e também pode-se observar na Figura 35, os modelos criados com as meta-arquiteturas R-FCN e SSD são rápidos e apresentam uma boa precisão. Por outro lado, o *Faster* R-CNN obteve as melhores precisões, porém trata-se de modelos lentos.

Tanto os resultados apresentados pelos autores da API quanto a possibilidade de construção de modelos rápidos e eficientes, reforçaram mais ainda a necessidade de trabalhar com uma plataforma unificada de redes neurais para detecção de objetos.

Entretanto, neste trabalho a avaliação de resultado dos modelos focou apenas na precisão das meta-arquiteturas, pois conforme será explicado nas sessões 4.1.4 e 4.1.5 o tempo de processamento e uso de memória está limitado aos recursos computacionais disponíveis para o desenvolvimento deste trabalho.

4.1.1 Base de dados

A tarefa inicial para treinar o detector de objetos consiste em montar uma base de dados de imagens que represente o universo de objetos que se quer identificar. Para atender a esta demanda, buscou-se por repositórios de dados ambientais que permitisse acessar imagens de cetáceos avistados em seu habitat natural. A busca nos levou aos conjuntos de dados de avistagens do

iNaturalist e dos dados de monitoramento de praias do Sistema de Informação de Monitoramento da Biota Aquática (SIMBA) para o PMP-BS.

O iNaturalist é um projeto de ciência cidadã que motiva a colaboração de entusiastas da natureza, ao registrar avistagens das mais diferentes espécies existentes em nosso planeta (INATURALIST, 2018). O repositório de dados do iNaturalist permite o registro de avistagens contendo a localização das ocorrências, descrição taxonômica de cada espécie e fotografias tiradas pelos usuários da plataforma. A vantagem de adotar as imagens provenientes do iNaturalist foi a diversidade de espécies encontradas no repositório de dados, bem como a garantia de validação dos registros por pesquisadores e especialistas da área.

O SIMBA é voltado a gestão das informações coletadas para o PMP-BS, e possibilita o registro de dados de monitoramento, ocorrências de fauna alvo do projeto, também contempla a manutenção dos dados de reabilitação dos animais encontrados vivos durante as atividades rotineiras de praia. Todo animal encontrado vivo ou morto durante o monitoramento recebe um cadastro no sistema, onde são inseridas fotografias tiradas pelas equipes de campo durante o processo de registro do indivíduo.

Diferente do iNaturalist cuja maioria das imagens apresentam indivíduos inseridos em ambiente aquático, as imagens do PMP-BS retratam em sua maioria, cenas de animais encontrados mortos nas praias, representando cenários de característica ambiental arenoso. Optou-se por utilizar as imagens do SIMBA, por se tratar de avistagens de animais recorrentes ao litoral brasileiro, além de possibilitar o enriquecimento de detalhes através da diversidade de cenários onde as fotografias foram obtidas.

Em ambos os repositórios se efetuou a pesquisa de dados visando encontrar registros de avistagens para animais da ordem dos cetáceos, que apresentassem fotografias vinculadas aos registros. A consulta foi realizada no dia 17 de dezembro de 2018, onde obteve-se o resultado de 8111 imagens provenientes do iNaturalist e 13822 imagens no SIMBA.

4.1.2 Seleção das imagens

Como o principal objetivo da etapa de detecção de objetos para este trabalho é localizar as dorsais dos indivíduos, foi necessário filtrar as imagens obtidas para atender a este escopo. No

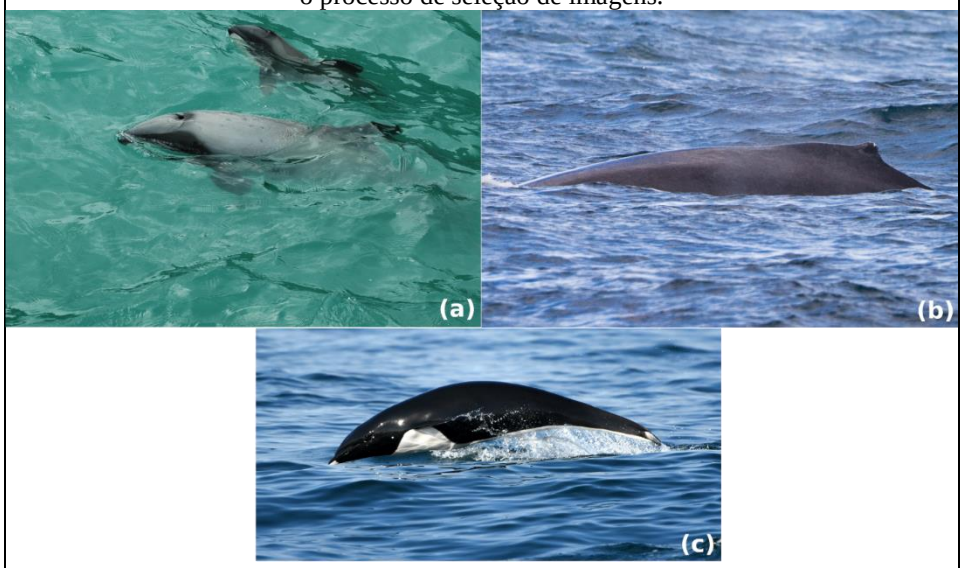
intuito de montar uma base de dados consistente para o treinamento de uma rede neural de detecção de objetos, foram criadas regras para exclusão das imagens que não atendiam ao propósito.

O primeiro critério consiste em excluir todas as imagens que não apresentem ao menos um indivíduo com a área de interesse visível, neste caso a dorsal. Portanto, foram excluídas imagens onde a dorsal encontrava-se oclusa (Figura 36a), indivíduos da ordem dos cetáceos cuja espécies não possuem nadadeira dorsal (Figura 36c), ou que não apresentem uma nadadeira relevante a ponto de retratar um formato distinguível pelo processo de treinamento da rede neural (Figura 36b).

O segundo critério de exclusão foi criado para atender uma demanda das imagens obtidas através do SIMBA. Como a maioria das imagens obtidas são de animais mortos, foi necessário excluir as imagens cujo indivíduo apresentava-se em um estado de decomposição avançado (Figura 37b), mesmo que a nadadeira dorsal estivesse visível.

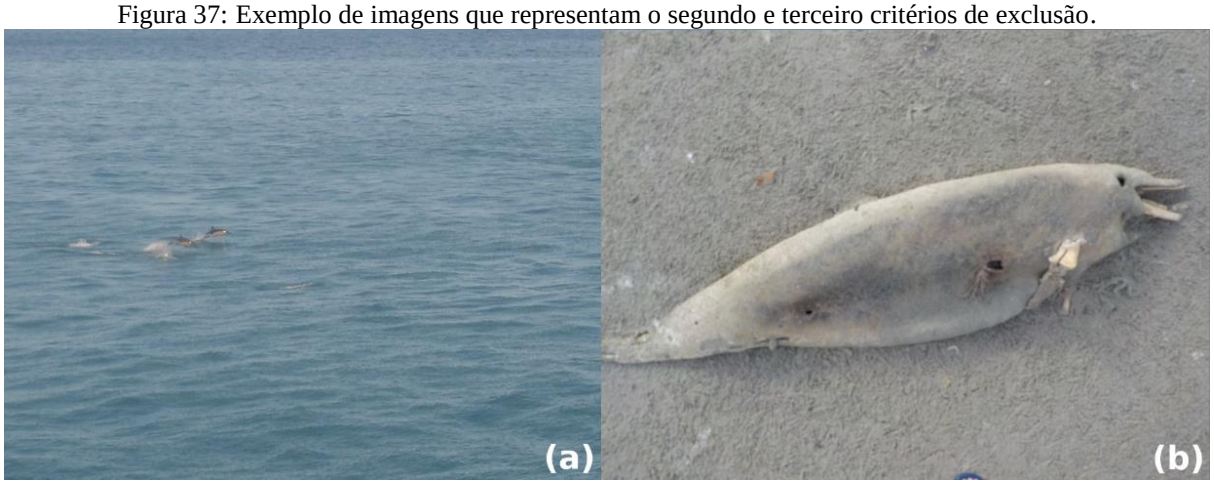
Já o terceiro e último critério de exclusão consiste na percepção visual empírica da pessoa que está selecionando as imagens, ao relacionar distância aparente entre a câmera e indivíduo na cena (Figura 37a). Ou seja, se a pessoa que estiver avaliando a imagem identificar que a representação do animal é pequena em relação ao tamanho total da imagem ou ao contexto da cena presenciada, esta imagem deve ser removida da base de dados.

Figura 36: Exemplo de imagens que representam os critérios de exclusão definidos para o processo de seleção de imagens.



Fonte: Adaptado de iNaturalist (2018).

Figura 37: Exemplo de imagens que representam o segundo e terceiro critérios de exclusão.



Fonte: Adaptado de iNaturalist (2018); PMP-BS (2017).

Nos casos de imagens que apresentam mais de um indivíduo por imagem, se ao menos um indivíduo passa por todos os critérios de exclusão o arquivo permanece na base de dados.

Após a avaliação manual das imagens, um total de 1913 imagens foram selecionadas, sendo 1489 do iNaturalist e 424 do SIMBA. A grande redução no número de imagens resultantes do SIMBA, decorreu-se pelo fato de que muitas imagens vinculadas aos registros representavam cenas das atividades de resgate dos animais, ou seja, em alguns casos não mostravam o animal ou apresentavam outras partes do corpo do indivíduo como, cabeça, cauda, etc.

4.1.3 Delimitação dos objetos de interesse

O treinamento dos modelos de redes neurais da API exige que seja informado a região da imagem que contém o objeto de interesse, ou seja, delimitar a região utilizando caixas delimitadoras, também conhecido pelo termo em inglês *bounding box*.

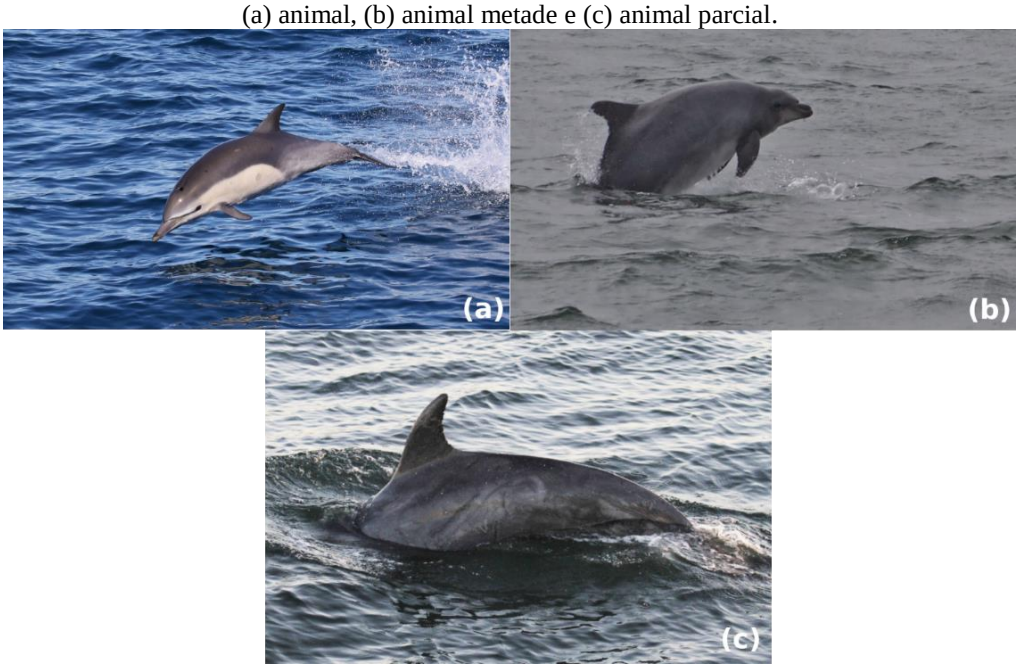
Para esta tarefa utilizou-se o software LabelImg (2015), que dispõem de uma ferramenta para delimitar as regiões dos objetos nas imagens, além de possibilitar que estes sejam rotulados. Também é possível salvar arquivos contendo as marcações e rótulos no padrão Extensible Markup Language (XML) para o modelo de anotações do PASCAL VOC (EVERINGHAM et al., 2010).

Inicialmente, a marcação das áreas de interesse focou no objeto alvo, ou seja, a dorsal. Posteriormente decidiu-se ampliar de forma exploratória o escopo do treinamento visando avaliar o desempenho da API, ao incluir alguns exemplos que representassem os animais em seu habitat.

Portanto, além de delimitar e rotular as regiões contendo as dorsais dos indivíduos nas imagens, foram criadas algumas caixas delimitadoras para identificar os indivíduos nas imagens, bem como foi definido três rótulos de identificação. Sendo estes:

- Animal: para identificar indivíduos com a maior parcela do corpo visível, por exemplo, animais fora da água, fotografias subaquáticas ou de animais encalhados na areia (Figura 38a);
- Animal metade: animais cuja parte frontal e dorsal estão visíveis, e o restante do corpo encontra-se submerso ou ocluso na imagem (Figura 38b);
- Animal parcial: animais parcialmente visíveis fora da água, na maioria dos casos trata-se de imagens que apresentam a dorsal e parte das costas dos indivíduos (Figura 38c).

Figura 38: Exemplo de imagens rotuladas para o treinamento de detecção de objetos. Rótulos: (a) animal, (b) animal metade e (c) animal parcial.



Fonte: Adaptado de iNaturalist (2018).

4.1.4 Modelo pré-treinado de rede neural

A transferência de aprendizado em redes neurais é um recurso que pode melhorar o desempenho da generalização de um modelo para uma nova tarefa (YOSINSKI et al., 2014), e consequentemente auxilia em casos de recursos computacionais limitados, tal como contribui em situações restritivas de tempo de processamento e conhecimento técnico (GARCIA-GASULLA et al., 2017).

Apoiando-se nessas premissas, fez-se uso de modelos pré-treinados durante a tarefa treinamento com a base de dados construída para este trabalho, ao invés de abordar o método de inicialização de pesos aleatórios para o treinamento das redes neurais.

Os idealizadores da API de detecção de objetos descrevem em seu artigo (HUANG et al., 2016), que o desenvolvimento desta focou em criar um *framework* que permita explorar os fatores de tempo, desempenho e acurácia durante a criação dos modelos pré-treinados, para que atuem em novas tarefas.

A Tabela 4 apresenta os modelos de redes neurais pré-treinados disponibilizados pela API e as avaliações obtidas durante o treinamento destes com a base de dados COCO (LIN et al., 2014). Para este trabalho, definiu-se que o critério de seleção dos modelos a serem utilizados focaria nos que apresentassem o melhor resultado de avaliação mAP para cada tipo de meta-arquitetura de rede neural, portanto foram escolhidos os modelos 7, 15 e 20.

Tabela 4. Comparativo de acurácia e velocidade de processamento para as meta-arquiteturas de redes neurais da API de detecção de objetos, utilizando a base de dados COCO e caixas delimitadoras.

Nº	Nome do modelo	Velocidade (ms)	COCO mAP[^1]
1	ssd_mobilenet_v1_coco	30	21
2	ssd_mobilenet_v1_0.75_depth_coco	26	18
3	ssd_mobilenet_v1_quantized_coco	29	18
4	ssd_mobilenet_v1_0.75_depth_quantized_coco	29	16
5	ssd_mobilenet_v1_ppn_coco	26	20
6	ssd_mobilenet_v1_fpn_coco	56	32
7	ssd_resnet_50_fpn_coco	76	35
8	ssd_mobilenet_v2_coco	31	22
9	ssd_mobilenet_v2_quantized_coco	29	22
10	ssdlite_mobilenet_v2_coco	27	22
11	ssd_inception_v2_coco	42	24
12	faster_rcnn_inception_v2_coco	58	28
13	faster_rcnn_resnet50_coco	89	30
14	faster_rcnn_resnet50_lowproposals_coco	64	-
15	rfcn_resnet101_coco	92	30
16	faster_rcnn_resnet101_coco	106	32
17	faster_rcnn_resnet101_lowproposals_coco	82	-
18	faster_rcnn_inception_resnet_v2_atrous_coco	620	37
19	faster_rcnn_inception_resnet_v2_atrous_lowproposals_coco	241	-
20	faster_rcnn_nas	1833	43
21	faster_rcnn_nas_lowproposals_coco	540	-

Fonte: Adaptado de Tensorflow Object Detection API (2017).

Cada modelo dispõe dos arquivos binários com as respectivas redes neurais já treinadas, bem como um arquivo com as configurações utilizadas durante o treinamento. Os APÊNDICES A, B e C demonstram os tipos de arquivos de configuração disponibilizados junto aos modelos e utilizados para o treinamento do modelo de detecção proposto para este trabalho.

Os parâmetros de configuração da rede neural podem ser alterados conforme a necessidade do problema. Porém, para o problema deste trabalho foi necessário alterar apenas algumas informações. O primeiro parâmetro a ser substituído é o número de classes de objetos a serem identificados, originalmente o número de classes existente na base de dados COCO é de 90, para

este trabalho será necessário identificar apenas 4 classes. O segundo parâmetro é o número de exemplos para validação do treinamento, que em nosso caso trata-se de 20% das imagens da base de dados, ou seja, 383 arquivos. Os demais parâmetros alterados restringem-se a localização em disco dos arquivos binários contendo a base de dados do trabalho, o modelo pré-treinado e os rótulos de identificação dos objetos.

O único modelo que foi necessário uma alteração nos parâmetros originais foi o modelo de número 20 descrito na Tabela 4. Foi necessário o ajuste do parâmetro de configuração de redimensionamento da imagem, alterando-o de 1200x1200 para 1024x768, esta alteração buscou reduzir o tempo de processamento ao minimizar o tamanho da matriz de dados processados. Os demais modelos mantiveram os parâmetros de configurações originais definidos pelos autores da API.

4.1.5 Treinamento

O treinamento das redes neurais foi executado em uma máquina com processador de 32 núcleos Intel Xeon CPU E5-2400 de 1.9GHz e 64GB de memória. O sistema operacional instalado é o Centos 7 com a versão 1.12 do Tensorflow.

Por se tratar de uma máquina onde processador não foi construído apenas para processamento de dados matriciais, como é o caso da *Graphics Processing Unit* (GPU). O tempo de treinamento aumenta consideravelmente conforme pode ser observado na comparação de tempo de processamento da Tabela 5.

E como o tempo para execução do treinamento e avaliação dos modelos era um fator crucial para obter os resultados desta etapa do trabalho, limitou-se a quantidade de combinações de treinamento para apenas os três modelos descritos na seção anterior.

Tabela 5. Comparativo de tempo de processamento por passo em milissegundos. Primeiro os tempos obtidos pelos autores da API utilizando uma GPU, na sequência o tempo alcançado neste trabalho utilizando um CPU de 32 núcleos.

Nome do modelo	**Tempo (ms) / passo GPU Nvidia GeForce GTX TITAN X**	**Tempo (ms) / passo CPU Intel Xeon E5-2400 de 1.9GHz**
ssd_resnet_50_fpn_coco	76	19000
rfcn_resnet101_coco	92	5000
faster_rcnn_nas	1833	27000

Por padrão, os arquivos de configuração das redes neurais estão programados para rodar em média 200 mil passos de treinamento para os modelos *Faster* R-CNN e R-FCN e 25 mil passos para o modelo SSD usando o banco de dados COCO. No entanto, devido ao longo tempo necessário para processar todas estas iterações, decidiu-se que o treinamento seria finalizado em mais ou menos 35 mil passos para os modelos *Faster* R-CNN e R-FCN e 4500 passos para o modelo SSD.

Levando em consideração que o tamanho de lotes de treinamento está configurado em um para os modelos *Faster* R-CNN e R-FCN, ou seja, cada passo do treinamento equivale ao processamento de apenas uma imagem por vez. E observando que a parcela de imagens destinadas ao treinamento é de 1530 arquivos, obtivemos mais ou menos 22 épocas de treinamento para cada modelo de rede neural, ou seja, cada imagem foi processada no mínimo 22 vezes.

Já para o caso do modelo SSD o tamanho de lote é equivalente a 32 imagens por passo necessitando algo em torno de 47,81 passos para processar uma época de treinamento, considerando o número total de passos treinados equivale a aproximadamente 94 épocas treinadas. As informações referentes aos tempos de processamento e números referentes a quantidade de passos e épocas de treinamento estão descritas na Tabela 6.

Tabela 6. Números relacionados ao processo de treinamento dos modelos pré-treinados escolhidos para este trabalho.

Nome do modelo	**Tempo (ms) / passo**	**Tempo total de processamento (hrs)**	**Nº total de passos**	**Nº total de épocas**
ssd_resnet_50_fpn_coco	168000	210	4500	94,12
rfcn_resnet101_coco	5000	48	34000	22,22
faster_rcnn_nas	27000	258	34349	22,45

4.2 SEGMENTAÇÃO

A etapa de segmentação é executada em paralelo com a etapa de detecção de objetos e consiste em separar a região de interesse do restante da imagem. Ou seja, separar o objeto em primeiro plano (*foreground*) do fundo (*background*). Em seguida extrai-se o contorno remanescente do objeto segmentado. No texto a seguir será retratado todas as etapas do desenvolvimento que envolvem esta tarefa, também descreveremos as técnicas envolvidas justificando a escolha das mesmas.

4.2.1 Segmentação do objeto de interesse

Conforme pode-se observar no Capítulo 3 alguns dos trabalhos relacionados fazem uso de métodos clássicos de segmentação (HALE, 2008; ANDREOTTI et al., 2017; HILLMAN et al., 2002), outros autores mesclam estas técnicas em uma única aplicação (CARVAJAL-GÁMEZ et al., 2017). Também houvena trabalhos que propuseram uma abordagem inovadora ao adotar o processo de segmentação utilizando CNN (WEIDEMAN et al., 2017; HUGHES; BURGHARDT, 2016). Contudo, estes métodos de segmentação não são capazes de descrever com precisão os limites que separam o *foreground* do *background*, deixando na maior parte dos casos um conteúdo residual nas regiões segmentadas.

Portanto, neste trabalho adotamos o método de segmentação semântica, que permite classificar cada pixel da imagem como *foreground* e *background* (GUO et al., 2018). Contudo, é válido salientar que este método apresenta duas limitações. Primeiramente, ao contrário dos métodos clássicos que simplificam a lógica utilizando fórmulas matemáticas, a segmentação semântica exige o treinamento de uma rede neural contendo as classes de objetos que deseja segmentar. A segunda limitação deste tipo de técnica de segmentação, é a incapacidade de separação de objetos sobrepostos em instâncias distintas.

A limitação referente a separação de objetos sobrepostos pode ser um problema nos casos de imagens que apresentem indivíduos aglomerados. No entanto, na maioria dos casos os pesquisadores que trabalham no controle populacional de cetáceos tendem a obter imagens individuais para cada animal da população, e também costumam separar antes as instâncias de cada indivíduo em imagens com múltiplos animais. Atitudes estas que viabilizam a adoção do método de segmentação semântica para este trabalho.

Outro fator relevante para a adoção de tal método, fica a cargo da ferramenta escolhida para esta tarefa também rodar para o *framework* TensorFlow, trata-se da ferramenta DeepLab. Além de ser uma ferramenta desenvolvida para a mesma plataforma utilizada na tarefa de detecção de objetos, o DeepLab pode ser considerado atualmente como o estado da arte dos modelos de aprendizagem profunda para segmentação semântica (CHEN et al., 2018), como pode ser constatado através dos resultados obtidos para as avalições da ferramenta no conjunto de dados de teste PASCAL VOC (2012) na Tabela 7 e Cityscapes (CORDTS et al., 2015) na Tabela 8.

Tabela 7. Comparação de resultados obtidos entre o DeepLab v3+ e os demais modelos de alta performance, na base de dados de teste do PASCAL VOC 2012.

Método	mIOU
Deep Layer Cascade (LC)	82.7
TuSimple	83.1
Large Kernel Matters	83.6
Multipath-RefineNet	84.2
ResNet-38 MS COCO	84.9
PSPNet	85.4
IDW-CNN	86.3
CASIA IVA SDN	86.6
DIS	86.8
DeepLabv3	85.7
DeepLabv3-JFT	86.9
DeepLabv3+ (Xception)	87.8
DeepLabv3+ (Xception-JFT)	89.0

Fonte: Adaptado de Chen et al. (2018).

Tabela 8. Comparação de resultados obtidos entre o DeepLab v3+ e os demais modelos de alta performance, na base de dados de teste Cityscapes com anotações de contorno grosseiras.

Método	Anotação grosseira	mIOU
ResNet-38	x	80.6
PSPNet	x	81.2
Mapillary	x	82.0
DeepLabv3	x	81.3
DeepLabv3+	x	82.1

Fonte: Adaptado de Chen et al. (2018).

Assim como no caso da detecção de objetos, o DeepLab dispõe de modelo pré-treinado para o treinamento de modelos específicos através da transferência de aprendizado. Portanto também fez-se uso do recurso para esta tarefa. No entanto, Chen et al. (2018) também adotaram o método de treinamento sem um modelo pré-treinado em seu trabalho, produzindo resultados significativos ao utilizar a base de dados PASCAL VOC. Deste modo, exploramos a mesma metodologia com a base de dados construída para este trabalho, com intuito de avaliar a eficiência entre os dois métodos.

Para realizar o treinamento da rede neural do DeepLab é necessário fornecer as imagens no formato RGB e as respectivas máscaras com os objetos segmentados e classificados por valores em escala de cinza em arquivo de imagem do tipo PNG sem a camada *alfa*. Sendo o valor 0 para identificar o background e os valores maiores que este para identificar cada classe de objetos.

Deste modo, foram anotados os segmentos dos indivíduos em 1359 arquivos do banco de dados das imagens provenientes do iNaturalist. O contorno foi feito utilizando a ferramenta *VGG Image Annotator* (VIA) (DUTTA; GUPTA; ZISSERMAN, 2016), que pode ser utilizada em navegadores para internet e permite delimitar os pontos de ligação do contorno dos objetos presentes na cena (Figura 39), as informações geradas são salvas no formato *JavaScript Object Notation* JSON.

As anotações dos segmentos foram classificadas como animal, animal metade e animal parcial, descartou-se a anotação para dorsal pois considerou-se que a mesma se trata de um segmento de contorno aberto e por sua vez poderia influenciar negativamente no treinamento do modelo. Utilizando a aleatoriedade para seleção de arquivos, foram separadas 951 imagens (70%) para o treinamento e 408 (30%) para testes e avaliação do modelo.

Figura 39: Interface do software VIA, para anotação de segmentos.

Fonte: Compilação do autor.

Inicialmente o treinamento com o modelo pré-treinado foi executado na mesma máquina descrita na seção 4.1.5 , no entanto o tempo demandado para a tarefa e o alto consumo de recurso de memória fez repensar sobre a influência do tamanho dimensional e em disco das imagens durante o processo. Portanto, optou-se por redimensionar as imagens baseando-se na área ocupada pelos indivíduos na cena.

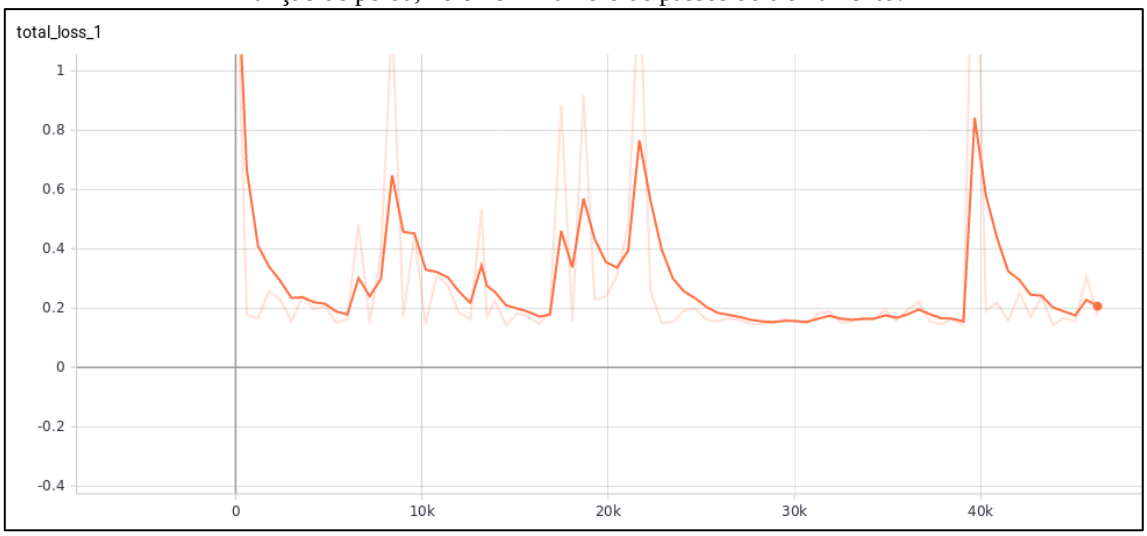
Para redimensionar as imagens avaliou-se os pontos que delimitam a anotação de contorno da máscara para cada indivíduo na imagem, visando encontrar as extremidades e executar o recorte da mesma. Nos casos em que imagem apresenta vários indivíduos as áreas eram somadas para encontrar a região que abrange todas as anotações. Posteriormente era acrescentado uma margem extra de 30 pixels para cada lado da área delimitada e então efetuava-se o recorte da imagem.

Este ajuste na dimensão da imagem, permitiu realizar o treinamento do modelo em uma máquina com GPU sem quebra de processo por falta de recurso de memória, reduzindo o tempo de processamento de 5 segundos por passo para 3.5 segundos por passo e o consumo de memória de 18GB para 7GB. A máquina utilizada para o treinamento possui um processador Intel I7 com quatro núcleos e 16GB de memória e uma placa de vídeo GTX745 com 384 núcleos CUDA[11] e 4GB de memória.

O treinamento com modelo pré-treinado utilizou as configurações recomendadas na documentação da ferramenta, conforme descrito no APÊNDICE D. O processo rodou durante 46875 passos e como o tamanho do lote estava definido em 1 foram executados algo em torno de 49 ciclos de treinamento, sendo finalizado ao ser observado que o valor da função de perda não apresentava uma alteração significativa de aprendizagem do modelo, como se pode observar através do gráfico da Figura 40.

[11] Abreviação para *Compute Unified Device Architecture*.

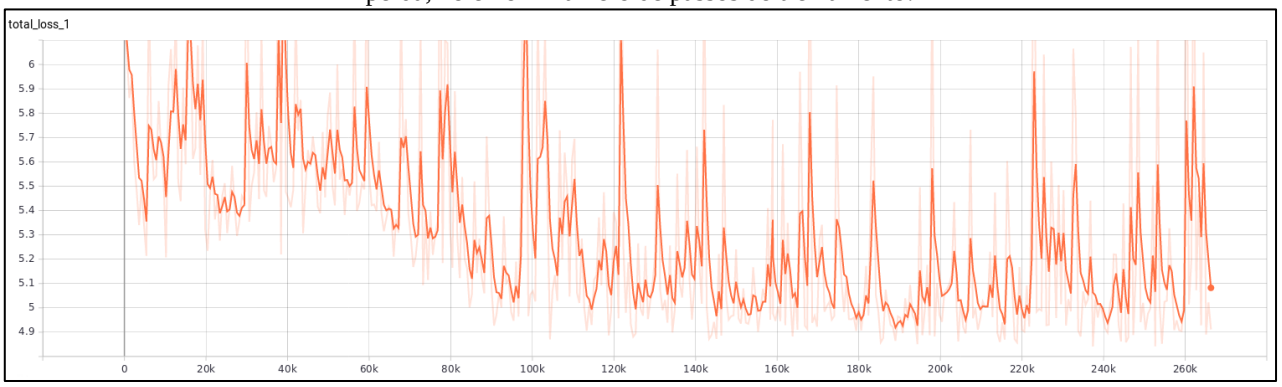
Figura 40: Gráfico da função de perda durante o treinamento com modelo pré-treinado. No eixo Y valor da função de perda, no eixo X número de passos do treinamento.



Fonte: Compilação do autor.

No caso do treinamento sem o modelo pré-treinado, como os idealizadores do DeepLab não descrevem em seu trabalho o tempo de treinamento nem o número de passos rodados, executou-se o treinamento da segmentação com as classes definidas por aproximadamente 366 mil passos, que para o nosso conjunto de dados contendo 951 imagens equivale a algo em torno de 385 ciclos de treinamento. O processo foi abordado ao observar que o modelo não apresentava uma melhora significativa na função de perda, conforme pode ser observado na Figura 41.

Figura 41: Gráfico da função de perda durante o treinamento sem o modelo pré-treinado. No eixo Y valor da função de perda, no eixo X número de passos do treinamento.



Fonte: Compilação do autor.

Conforme será apresentado no capítulo 5 o treinamento sem o modelo pré-treinado não obteve resultados significativos que justifiquem o uso deste na etapa de extração da linha de

contorno da dorsal, portanto o mesmo foi desconsiderado para a definição do processo deste trabalho.

4.3 EXTRAÇÃO DA LINHA DE CONTORNO DA DORSAL

Apesar da segmentação semântica apresentar bons resultados ao gerar uma máscara do objeto de interesse, está máscara não fornece um bom detalhamento da dorsal. Por este motivo, fez-se uso do método de refinamento abordado por Hughes e Burghardt (2016), trata-se dos algoritmos para resolver o problema de fosqueamento do termo inglês *matting* ou *digital image matting*.

Em um trabalho preliminar dos autores (HUGHES; BURGHARDT, 2015), foram avaliados três algoritmos *matting*, o *affinity matting* e *colour matting* descrito por Zheng e Kambhamettu (2009) e *GrabCut* por Rother, Kolmogorov e Blake, (2004). Os testes foram efetuados em 120 imagens de dorsais cujo contorno foi desenhado manualmente para comparar com os resultados obtidos pelos algoritmos (Tabela 9). Ao finalizar os testes os autores concluíram que o melhor algoritmo para o problema de reconstrução de contorno da dorsal, devido ao auto índice de precisão é o *affinity matting*.

Tabela 9. Comparação de resultados obtidos para os algoritmos *matting* no trabalho de Hughes e Burghardt (2015).

Método	**Precisão (s) (pixels)**	**Robustez (s=0.016)(pixels)**	**Tempo de processamento (s=0.016)(segundos)**
Affinity matting	0.877 (0.009)	1.001	64.43
Colour matting	1.970 (0.005)	>2.454	>302.5
GrabCut	1.366 (0.006)	1.9431	9.87

Fonte: Adaptado de Hughes e Burghardt (2015).

Os valores de precisão apresentados na segunda coluna da Tabela 9, definem a avaliação do menor erro produzido por um método ao comparar a linha desenhada a mão com o resultado obtido com algoritmo. Quanto menor o valor mais preciso é o algoritmo. Já a robustez apresentada na terceira coluna, é calculada como erro médio da localização em linhas de intersecção entre *foreground* e *background* com a maior espessura.

Apesar do trabalho de Hughes e Burghardt (2015) demostrar que o algoritmo *affinity matting* obteve os melhores resultados, considerou-se a possibilidade de explorar outros algoritmos descritos na literatura (SINGH; JALAL, 2013), visando avaliar o desempenho dos mesmos para o

problema de refinamento da linha de contorno da dorsal dos animais da ordem dos cetáceos. Logo escolheu-se os seguintes algoritmos:

1. *Learning Based* (ZHENG; KAMBHAMETTU, 2009);
2. *Bayesian* (CHUANG et al., 2001);
3. *Knn* (CHEN; LI; TANG, 2013);
4. *Closed form* (LEVIN; LISCHINSKI; WEISS, 2007);
5. *Lkm* (HE; SUN; TANG, 2010); e
6. *Ifm* (AKSOY; AYDIN; POLLEFEYS, 2017).

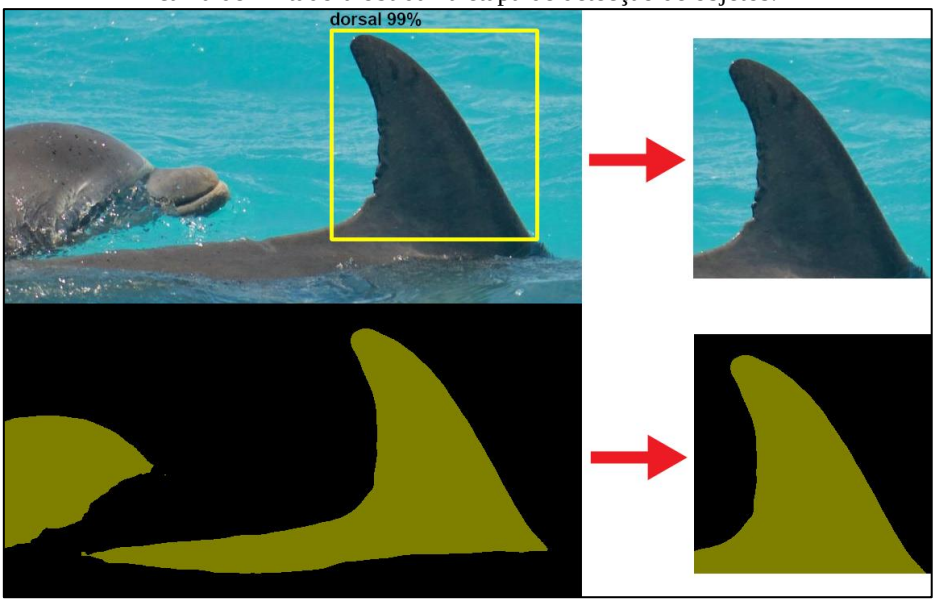
A escolha por estes algoritmos deu-se por tratar de implementações de código aberto em Python, linguagem de programação também utilizada para detecção de objetos e segmentação[12]. Também é válido ressaltar que o algoritmo 1 apesar de apresentar um nome diferente do que fora descrito por Hughes e Burghardt (2015) como *affinity matting*, trata-se do mesmo algoritmo que também foi implementado em um segundo trabalho de Hughes e Burghardt (2016), por este motivo o mesmo foi incluído a lista para comparar o seu desempenho perante aos demais algoritmos escolhidos.

Antes de aplicar o refinamento de contorno com os algoritmos *matting* selecionados, é necessário criar uma máscara da área de interesse contextualizando as regiões de *foreground*, *background* e área de intersecção. Processo este conhecido como *trimap*.

Para esta atividade, utiliza-se as caixas delimitadoras da dorsal encontradas durante a execução da etapa de detecção de objetos para recortar a região que representa o objeto na imagem resultante da tarefa de segmentação, conforme exemplificado na Figura 42. Porém antes de efetuar o recorte aplica-se uma margem extra de 10% em relação ao tamanho total da caixa delimitadora, para evitar o risco de perder alguma área da dorsal oclusa pela predição.

[12] Os algoritmos foram desenvolvidos e disponibilizados por Marco Forte em seu repositório de códigos (https://github.com/MarcoForte/closed-form-matting) e Aksoy, Aydın e Pollefeys (2017) no repositório de código (https://github.com/99991/matting).

Figura 42: Recorte da dorsal para imagem original e o respectivo segmento, usando a caixa delimitadora obtida na etapa de detecção de objetos.

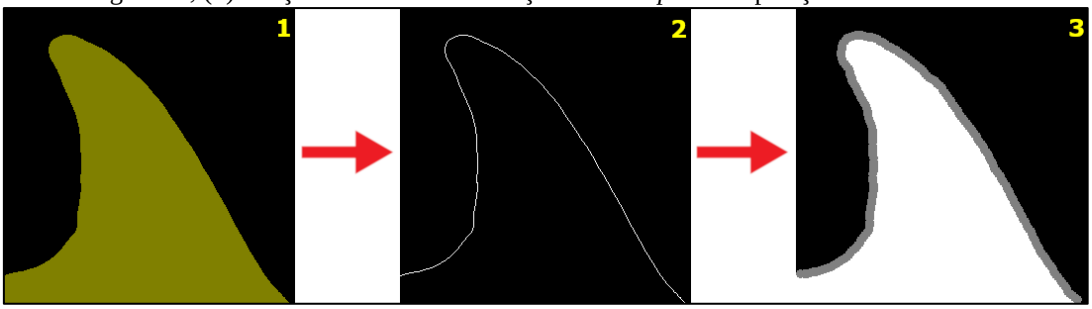


Fonte: Compilação do autor.

Após o recorte da imagem segmentada, extrai-se a linha de contorno do segmento (Figura 43 (2)), que deve ser dilata e sobreposta ao segmento novamente para delimitar a região de intersecção. O resultado é a máscara *trimap* com as cores branco (*foreground*), preto (*background*) e a cor cinza para a região que será analisada pelos algoritmos *matting* (Figura 43 (3)).

Considerando que em alguns segmentos das dorsais a linha extraída diverge da linha de interesse da dorsal, foi escolhido duas espessuras para a região de intersecção. As duas configurações de dilatação utilizam um *kernel* de 3x3 pixels com 2 e 3 interações, e conforme será apresentado no capítulo 5 passou por um processo de avaliação visual para definir a melhor configuração para o problema proposto.

Figura 43: Passos para criação do *trimap*. (1) recorte do segmento na região da dorsal; (2) extração da linha de contorno do segmento; (3) criação da área de intersecção do *trimap* e sobreposição desta no semento da dorsal.



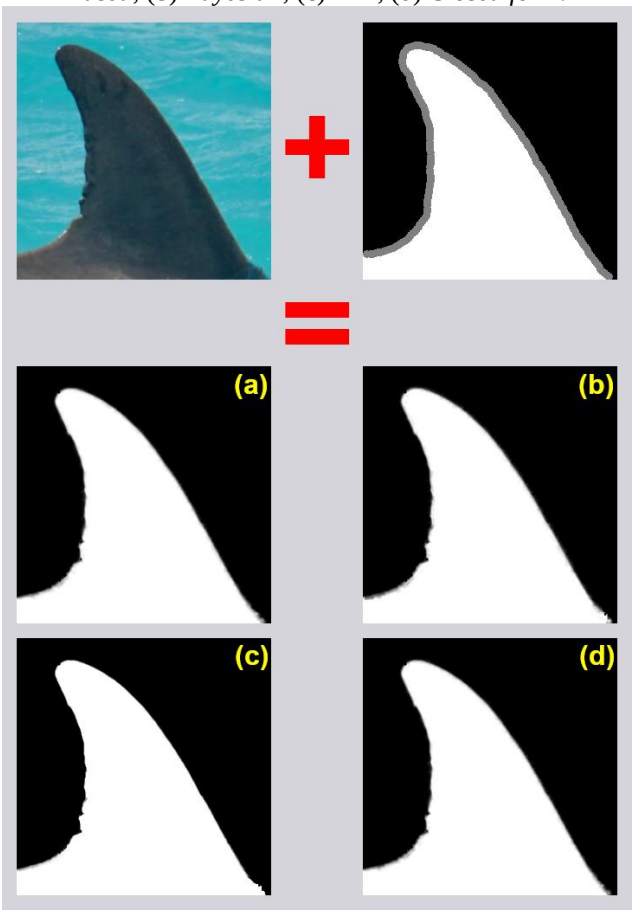
Fonte: Compilação do autor.

De posse da máscara**,** aplica-se o mesmo recorte da região da dorsal feito para a imagem de segmentação na imagem original no formato RGB. Estas então são processadas pelos algoritmos *matting*.

O resultado final do processo consiste em uma matriz de valores entre 0 e 1, semelhante uma camada *alpha* que descreve a intensidade dos pixels pertencentes ao conjunto de dados do *foreground* e *background*. A Figura 44 demonstra alguns exemplos de camada *alpha* resultante do processo de *matting*.

Com intuito de tornar visível a representação da área de intersecção entre a dorsal e o restante da cena na camada *alpha*, os valores entre 0 e 1 foram substituídos por valores da escala de cores de tons de cinza (0-255). Sendo o *foreground* representado pela cor branca 255, *background* cor preta 0 e a intersecção das regiões com os demais valores.

Figura 44: Exemplo dos resultados obtidos com os algoritmos *matting*. (a) *Learning Based*; (b) *Bayesian*; (c) *Knn*; (d) *Closed form*.

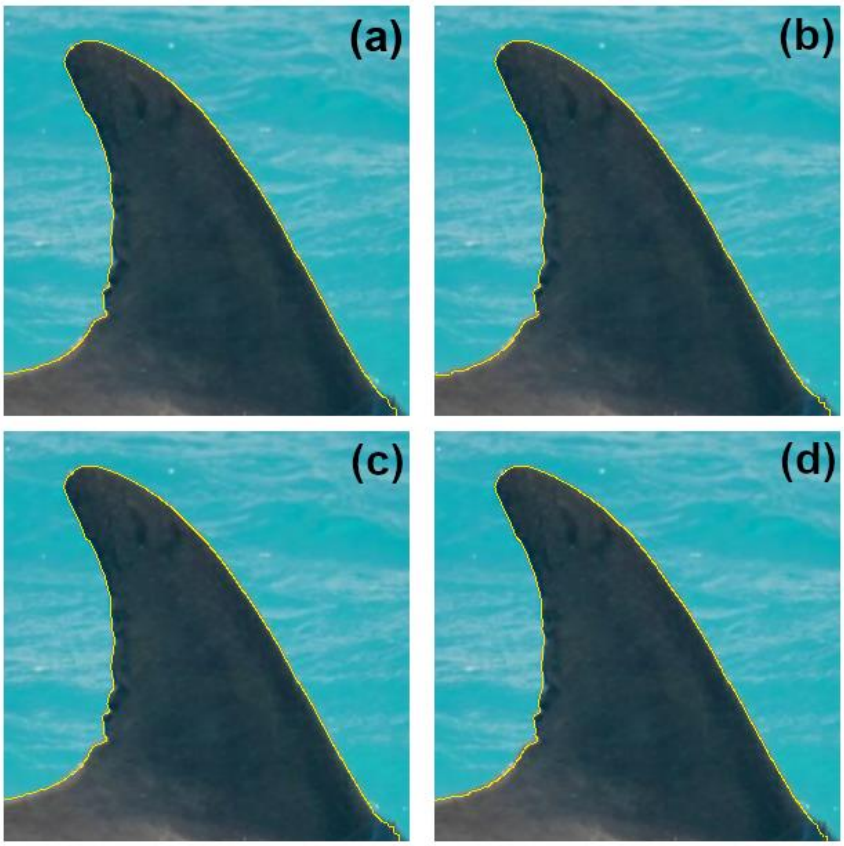


Fonte: Compilação do autor.

Todavia, para extrair a linha de contorno da área de intersecção é necessário transformar a máscara resultante do processo de *matting* em uma representação binária. Ou seja, deve-se delimitar um valor de corte que permita representar exatamente a área que representa a dorsal de um indivíduo e área que contextualiza o *background*. Hughes e Burghardt (2015) adotaram um valor de corte da camada *alpha* em 0,5, onde todo valor < 0,5 recebe o valor 0 e os valores >= 0,5 o valor 1. Neste trabalho, adotamos a metodologia proposta pelos autores, além de propormos um método a nível exploratório, que consiste em calcular a média ponderada dos valores da camada *alpha* para definir o limiar de corte para a binarização.

Após a binarização, a linha de contorno é extraída com o algoritmo de detecção de contornos Canny (1986). A Figura 45 apresenta alguns exemplos de resultados obtidos ao final da tarefa, sobrepondo a linha resultante da etapa de extração da linha de contorno da dorsal na imagem original.

Figura 45: Exemplos dos resultados obtidos na etapa de extração da linha de contorno da dorsal. (a) *Learning Based*; (b) *Bayesian*; (c) *Knn*; (d) *Closed form*.



Fonte: Compilação do autor.

5 RESULTADOS

5.1.1 Avaliação da etapa de detecção de objetos

A avaliação da etapa de detecção de objeto fez uso das 383 imagens da base de dados que foram separadas para testes e validação. As imagens contêm as anotações das caixas delimitadoras dos objetos de interesse, sendo 497 objetos do tipo dorsal, 131 animal, 67 animal parcial e 36 animal metade. Totalizando 695 objetos nas 383 imagens. Estes, por sua vez foram considerados como o padrão verdade para o processo de avaliação.

Para produzir os resultados foi utilizado a funcionalidade de avaliação disponibilizado pela API de detecção de objetos. O método implementado pela API segue as regras definidas pela métrica de avaliação para detecção de objetos COCO (2015). Os resultados obtidos estão presentes na Tabela 10.

Tabela 10. Resultados obtidos durante a avaliação da etapa de detecção de objetos.

Modelo	**AP$^{IoU=.50:.95}$**	**AP$^{IoU=.50}$**	**AP$^{IoU=.75}$**	**AP^{small}**	**AP^{medium}**	**AP^{large}**	**AR max=100**	**AR small**	**AR medium**	**AR large**
SSD	**0.547**	**0.693**	**0.649**	**0.463**	**0.629**	**0.555**	0.663	**0.550**	**0.687**	0.670
R-FCN	0.497	0.689	0.630	0.360	0.556	0.506	0.590	0.475	0.608	0.599
Faster R-CNN	0.523	0.672	0.636	0.383	0.602	0.532	**0.670**	0.500	0.683	**0.677**

Em um primeiro momento, ao examinar os resultados obtidos para cada modelo de rede neural escolhida, pode-se observar que o modelo de rede neural SSD supera os demais em quase todos os quesitos da avaliação, com exceção de **AR$^{max=100}$** e **AR^{large}**.

Estes resultados representam um panorama global para as quatro classes de objetos, além de fornecer indícios suficientes de que o modelo de rede neural SSD é a melhor escolha para a etapa de detecção de objetos. No entanto, é preciso detalhar a avaliação entre as classes de objetos para investigar se o bom desempenho do modelo se repete para a classe de objeto da dorsal.

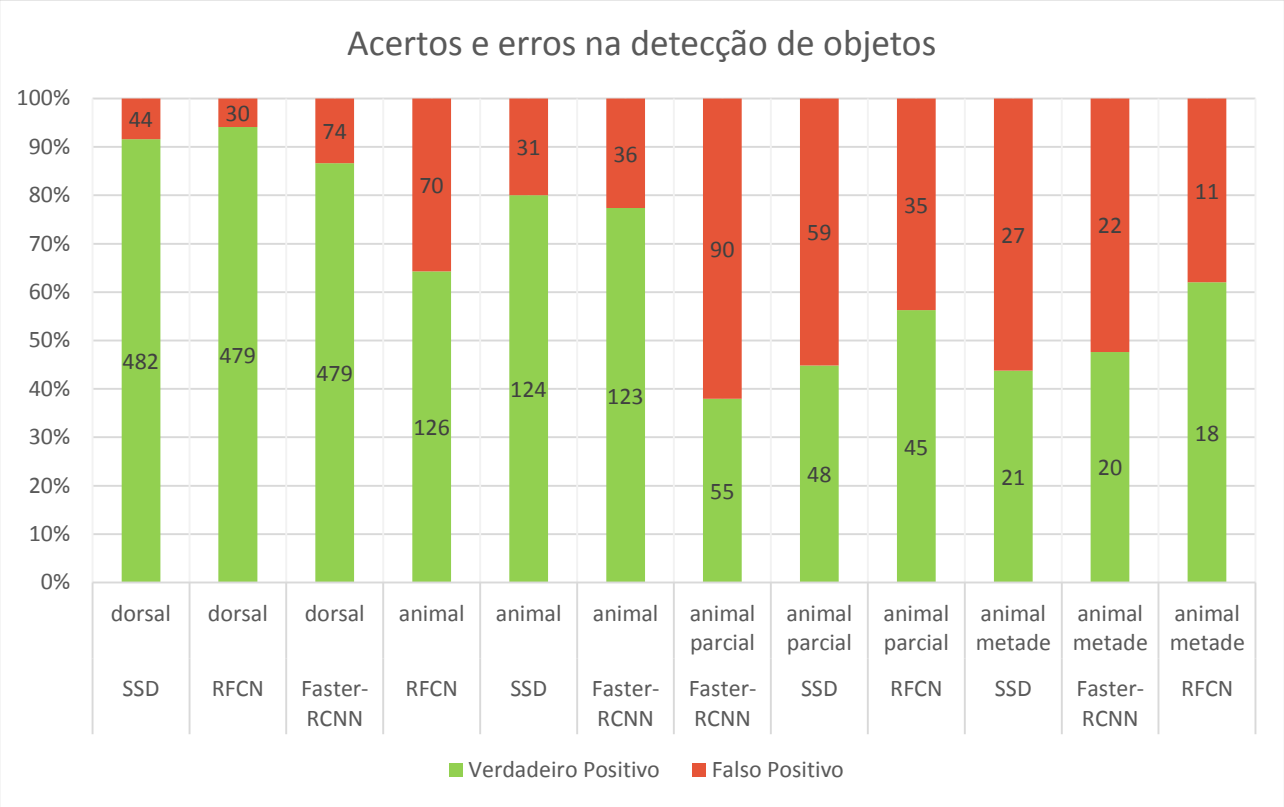
Para obter resultados detalhados do desempenho dos modelos para cada objeto detectado, aplicou-se a avaliação utilizando a métrica disponibilizada em PASCAL VOC (2012). Que permite avaliar o valor AP para cada classe de objetos, bem como calcula a média ponderada para todos os resultados individuais de AP do processo de detecção. O limiar de corte para considerar se um

objeto detectado é um verdadeiro positivo em relação ao padrão verdade definido pela métrica é IoU=.50. Os resultados obtidos são apresentados na Tabela 11 e Figura 46.

Tabela 11. Resultados obtidos para a avaliação da detecção de objetos com a métrica PASCAL VOC.

Modelo	**mAP %**	**AP % dorsal**	**AP % animal**	**AP % animal metade**	**AP % animal parcial**
SSD	**69,62**	**95,97**	90,85	**46,06**	45,61
R-FCN	68,86	95,62	**93,08**	42,73	44,00
Faster R-CNN	67,47	94,78	88,66	39,81	**46,62**

Figura 46: Gráfico com o número de acertos e erros encontrados durante a detecção de objetos, utilizado o IoU=0.5.



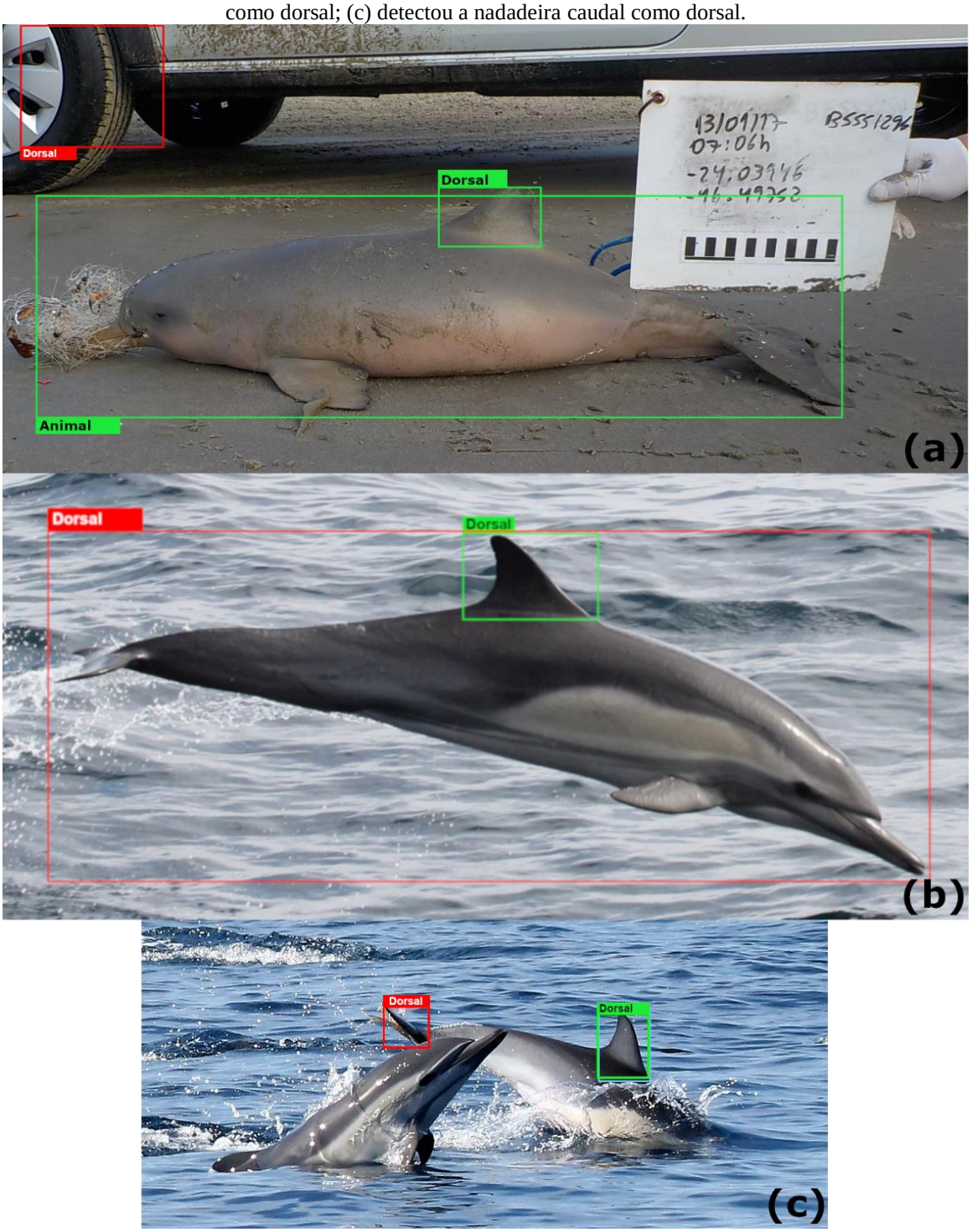
Fonte: Compilação do autor.

Observa-se que, novamente o modelo SSD destacou-se em relação aos demais modelos com o melhor resultado global mAP e também para a detecção de objetos do tipo dorsal e animal metade, porém foi superado pelo modelo R-FCN na classe de objetos animal e pelo modelo *Faster* R-CNN na classe animal parcial.

O gráfico da Figura 46 mostra a distribuição de acertos e erros para cada classe de objeto nos três modelos, no caso da dorsal o modelo SSD obteve o maior número de detecções positivas sendo apenas 44 destas consideradas falsas. Ao avaliar visualmente as detecções consideradas como falso positivo para este modelo, descobriu-se que apenas 18 eram efetivamente um erro conforme apresentado nos exemplos da Figura 47. Os 26 falsos positivos restantes eram dorsais que não foram inicialmente anotadas no padrão verdade por apresentarem as regiões de interesse parcialmente oclusas pela água ou por apresentarem uma região cuja quantidade de pixels é pequena (Figura 48), demostrando a eficiência do modelo para a tarefa proposta.

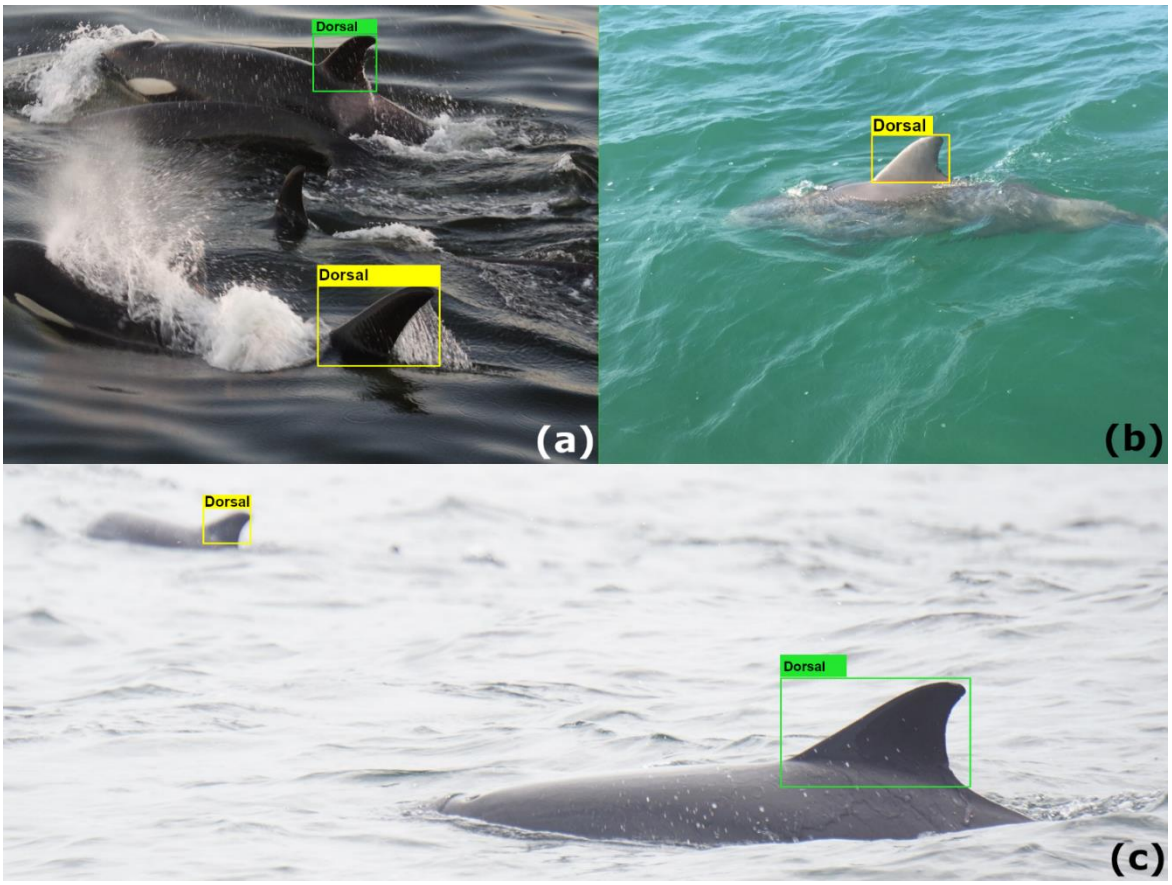
Apesar dos bons resultados obtidos pelo modelo SSD, pode-se denotar que os demais modelos avaliados apresentaram resultados aproximadamente equilibrados, confirmando o que fora dito por Yosinski et al. (2014) sobre a melhora de um determinado modelo ao fazer uso da transferência de aprendizado. Entretanto, a divisão dos animais parcialmente oclusos entre as classes animal metade e animal parcial geraram um desempenho abaixo do esperado, indicando uma possível limitação dos modelos em diferenciar os dois tipos de objetos.

Figura 47: Exemplos de erros de detecção gerados pelo modelo SSD destacados em vermelho, em verde das detecções corretas para os objetos do escopo. (a) detectou o pneu como dorsal; (b) detectou o animal como dorsal; (c) detectou a nadadeira caudal como dorsal.



Fonte: Compilação do autor.

Figura 48: Exemplos de detecção falso positivo gerados pelo modelo SSD destacados em amarelo, em verde as detecções corretas de dorsais. (a) detecção de dorsal parcialmente oclusa pela água; (b) detecção de dorsal não anotada no padrão verdade; (c) detecção de dorsal não anotado no padrão verdade devido ao seu tamanho e distância da câmera.



Fonte: Compilação do autor.

5.1.2 Avaliação da etapa de segmentação

Para avaliar a segmentação das três classes de objetos nas 408 imagens separadas para testes foi utilizado a métrica disponibilizada pelo próprio DeepLab, que resulta na média de todos os valores obtidos no cálculo de intersecção sobre a união da área segmentada com a área delimitada como padrão verdade (mIoU).

Para o treinamento da segmentação semântica utilizando o modelo pré-treinado, obteve-se o resultado mIoU de 70,3%, ficando apenas 11,8% abaixo do melhor resultado obtido pelos idealizadores do DeepLab ao utilizar uma base de dados de teste contendo 30 classes distintas e anotações grosseiras (CHEN et al., 2018). Já para o método de treinamento sem um modelo pré-treinado, o resultado obtido pela avaliação foi um mIoU de 36,68%, ou seja, 33,62% abaixo do valor obtido com o modelo pré-treinado e distante dos 87,8% obtidos pelos autores citados.

Estes resultados expõem a fragilidade do treinamento sem um modelo pré-treinado para a tarefa de segmentação, bem como reforça a necessidade de abordar a transferência de aprendizado para obter um resultado consistente e sem a necessidade de longos períodos de treinamento.

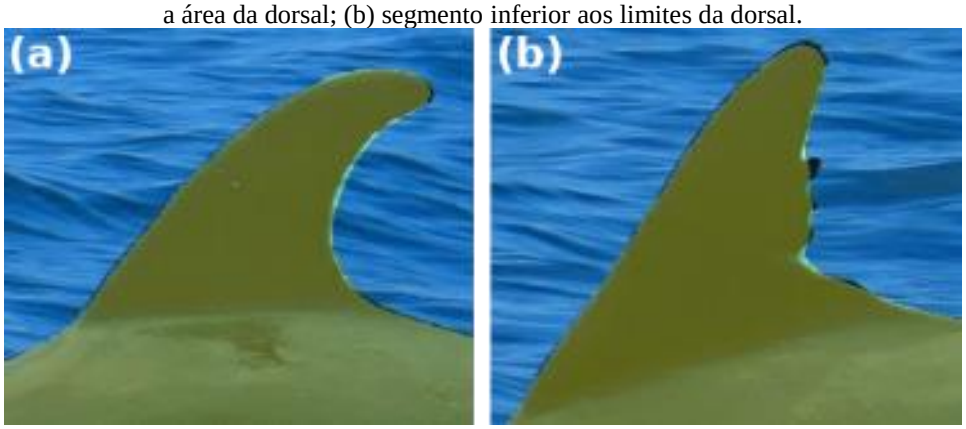
5.1.3 Análise visual da etapa de segmentação para criação do *trimap*

Levando em consideração que a avaliação da segmentação se restringe a um único valor global que representa a eficiência do modelo, decidiu-se analisar visualmente os segmentos gerados durante os testes sobrepondo-os as imagens originais, com intuito de observar a cobertura do segmento nas dorsais dos indivíduos e avaliar o potencial uso da etapa de segmentação para criação do recurso de *trimap*.

Das 408 imagens verificadas, 151 apresentaram algum tipo de inconsistência na região da dorsal que consequentemente influenciariam negativamente na criação do *trimap*. Observou-se em algumas dorsais a região segmentada extrapolava os limites entre a dorsal e o fundo (Figura 49a), em outros momentos o segmento era inferior aos limites, ou seja, ocupava uma área menor que a dorsal (Figura 49b).

Considerando o total de imagens inconsistentes, 81 apresentaram o excesso no segmento e 54 imagens a falta deste, resultando em um total de 33% de imagens que merecem atenção ao escolher espessura da região de intersecção do *trimap* para a etapa de extração da linha de contorno da dorsal. As 16 imagens restantes que não entraram nestas duas classificações são de dorsais que não foram segmentadas (Figura 50), e consequentemente foram descartadas.

Figura 49: Exemplos de inconsistências geradas pela segmentação. (a) segmento extrapola a área da dorsal; (b) segmento inferior aos limites da dorsal.



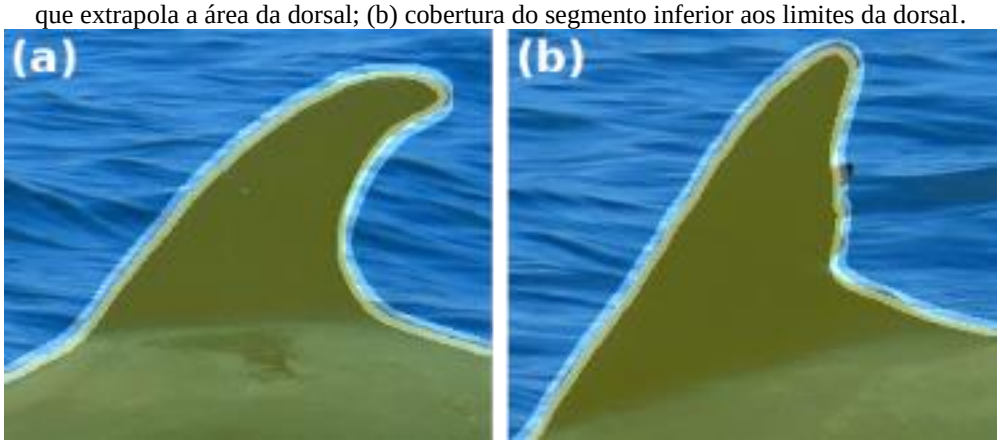
Fonte: Compilação do autor.

Figura 50: Indivíduo cuja dorsal não foi anexada a região segmentada.

Fonte: Compilação do autor.

Tendo em vista que na seção 4.3 definiu-se duas espessuras para a área de intersecção do *trimap*, avaliou-se quais destas seria capaz de recrutar o máximo de imagens inconsistentes para o processo de *matting*. Sobrepomos as dorsais originais com as duas configurações de espessuras geradas a partir da linha de contorno dos respectivos segmentos (Figura 51). Os resultados obtidos nessa avaliação podem ser observados na Tabela 12.

Figura 51: Exemplo de sobreposição com a área de intersecção. (a) cobertura do segmento que extrapola a área da dorsal; (b) cobertura do segmento inferior aos limites da dorsal.



Fonte: Compilação do autor.

Tabela 12. Número de imagens inconsistentes recrutadas para o processo de *matting*, após a avaliação visual das configurações de espessura da área de intersecção do *trimap*.

Configuração	Nº recrutamento inconsistência 1	Nº recrutamento inconsistência 2
kernel 3x3 2 interações	26	15
kernel 3x3 3 interações	66	34

Esta avaliação revelou que para a primeira configuração apenas 32% imagens com excesso de segmento poderiam ser recrutadas novamente, opostamente com a segunda configuração é possível recrutar 81% das imagens. Já no caso das imagens com segmento inferior a largura da dorsal, a primeira configuração recrutou apenas 28% das imagens e a segunda configuração 63%.

Compreendendo que a segunda configuração permite recrutar um número maior de imagens inconsistentes para o processo de *matting*, optou-se por utilizar esta como padrão para a etapa de avaliação de extração da linha de contorno descrita na próxima seção.

5.1.4 Avaliação da etapa de extração da linha de contorno da dorsal

A avaliação de desempenho das combinações criadas para o refinamento e extração da linha de contorno da dorsal, consiste basicamente em comparar os resultados obtidos com um conjunto de dados de contornos das dorsais desenhados a mão. O desenho manual define o padrão verdade a ser comparado e deve representar o contorno fino da dorsal com apenas um pixel de espessura.

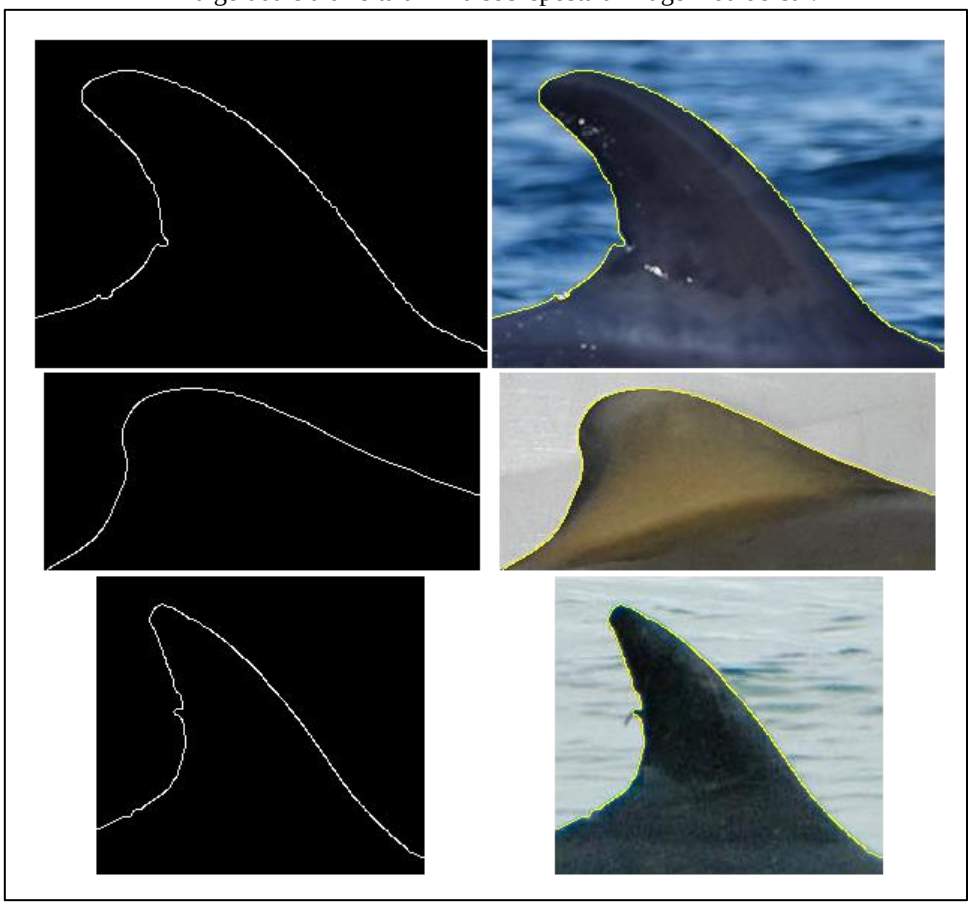
Durante a criação da base de dados para o treinamento da segmentação foram desenhados os contornos grosseiros tanto da dorsal quanto do animal. No entanto, estes não puderam ser utilizados aqui devido a descrição imprecisa de alguns detalhes, como por exemplo, os entalhes das dorsais. Neste caso, decidiu-se que um novo conjunto de dados contendo os contornos finos necessários para a avaliação seria criado do zero.

Como os trabalhos relacionados não descrevem qualquer tipo de método para seleção do conjunto de dados para avaliação, optou-se pela criação de um método que não envolve a seleção aleatória ou manual. Portanto, as imagens escolhidas para esta avaliação são provenientes da base de dados de teste da etapa de detecção de objetos. Como critério de seleção da dorsal, definiu-se que apenas as que alcançassem o valor superior ou igual de 75% IoU ao comparar caixa delimitadora da detecção de objetos e o padrão verdade, seriam escolhidas para produzir a linha de

contorno fino do padrão verdade. Das 383 imagens da base de testes, restaram 91 imagens com 98 dorsais que atendiam aos critérios.

Após a criação do contorno fino feito à mão, os mesmos arquivos contendo as dorsais selecionadas passaram pelo processo de segmentação e criação do *trimap*. Porém 10 destas geraram segmentos inconsistentes e por este motivo foram descartadas do conjunto de dados de avaliação. As dorsais restantes foram submetidas ao processo de *matting* e binarização com os limiares de corte 0,5 e média ponderada. A Figura 52, demonstra alguns exemplos de resultados gerados ao final do processo.

Figura 52: Resultados da etapa de extração da linha de contorno da dorsal, a esquerda a linha gerada e a direita a linha sobreposta a imagem da dorsal.



Fonte: Compilação do autor.

A métrica escolhida para avaliar os algoritmos faz uso dos conceitos definidos pela medida *F-Score* e foi descrita por Martin, Fowlkes e Malik (2004), que posteriormente foi incorporada por Arbelaez et al. (2010) como método de avaliação de detecção de contorno e segmentação, no centro de pesquisas em computação visual de Berkeley Universidade da Califórnia.

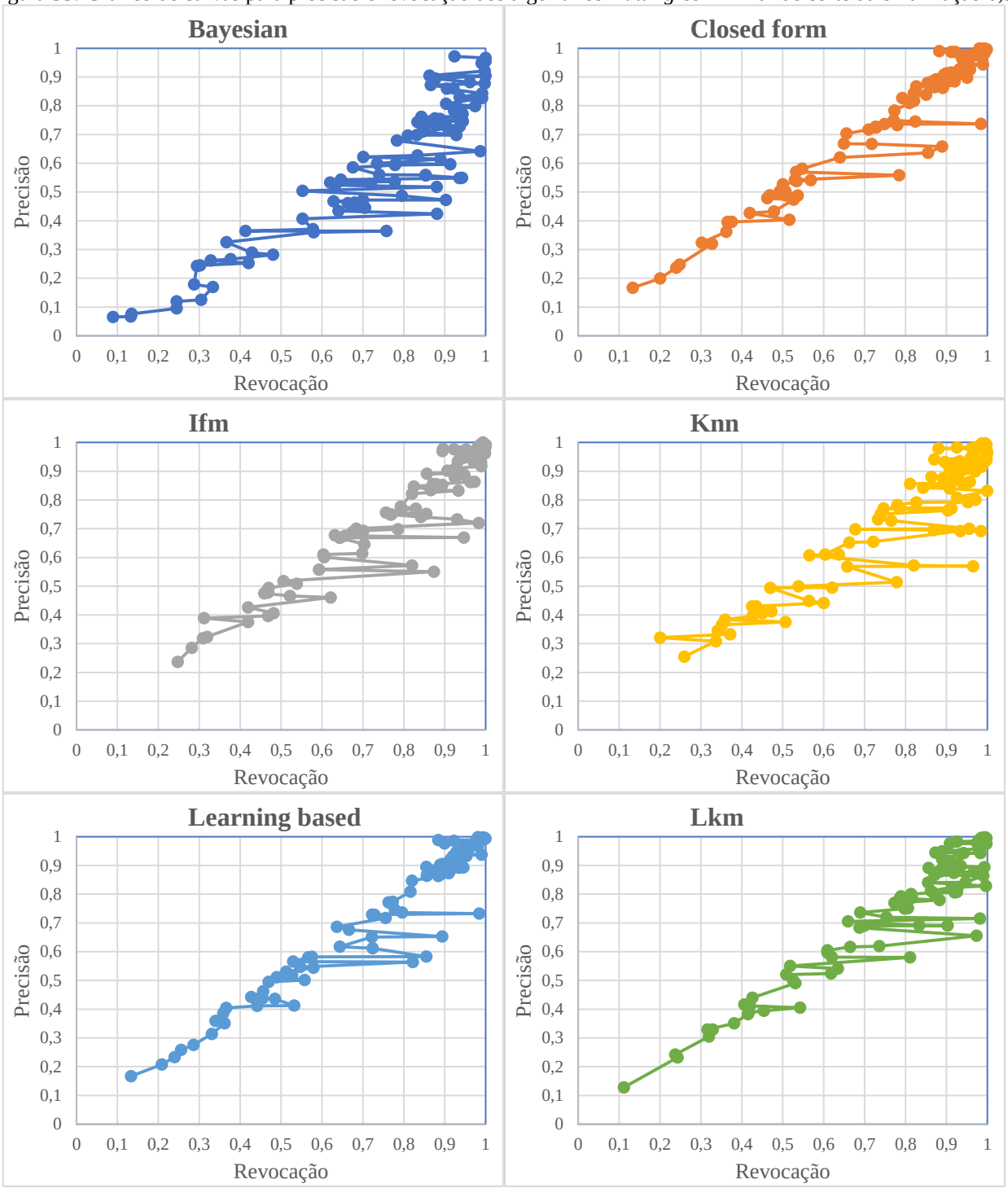
Os resultados globais obtidos durante a avaliação dos algoritmos com os respectivos limiares de corte da binarização são apresentados na Tabela 13. Observa-se que quase todos os algoritmos possuem uma variação de *F-Score* de até 2%, à exceção é o *Bayesian* que apresentou uma diferença maior que 8% com relação aos demais. Entretanto, em um primeiro momento, os valores de precisão e revocação indicam que em quase todas as combinações existe uma relação estreita entre a capacidade de encontrar e selecionar os pixels relevantes para a construção da linha de contorno da dorsal.

Tabela 13. Resultados globais para cada combinação de algoritmo e limiar de corte da binarização.

Algoritmo *matting*	**Limiar de corte binarização**	**Precisão**	**Revocação**	***F-score***
Ifm	média	0,838	0,878	0,858
Ifm	0,5	0,834	0,875	0,854
Knn	0,5	0,822	0,883	0,851
Knn	média	0,822	0,882	0,851
Lkm	0,5	0,827	0,869	0,848
Closed form	média	0,838	0,852	0,845
Lkm	média	0,822	0,869	0,845
Learning based	média	0,835	0,854	0,844
Closed form	0,5	0,833	0,847	0,840
Learning based	0,5	0,827	0,847	0,837
Bayesian	média	0,675	0,848	0,752
Bayesian	0,5	0,675	0,846	0,751

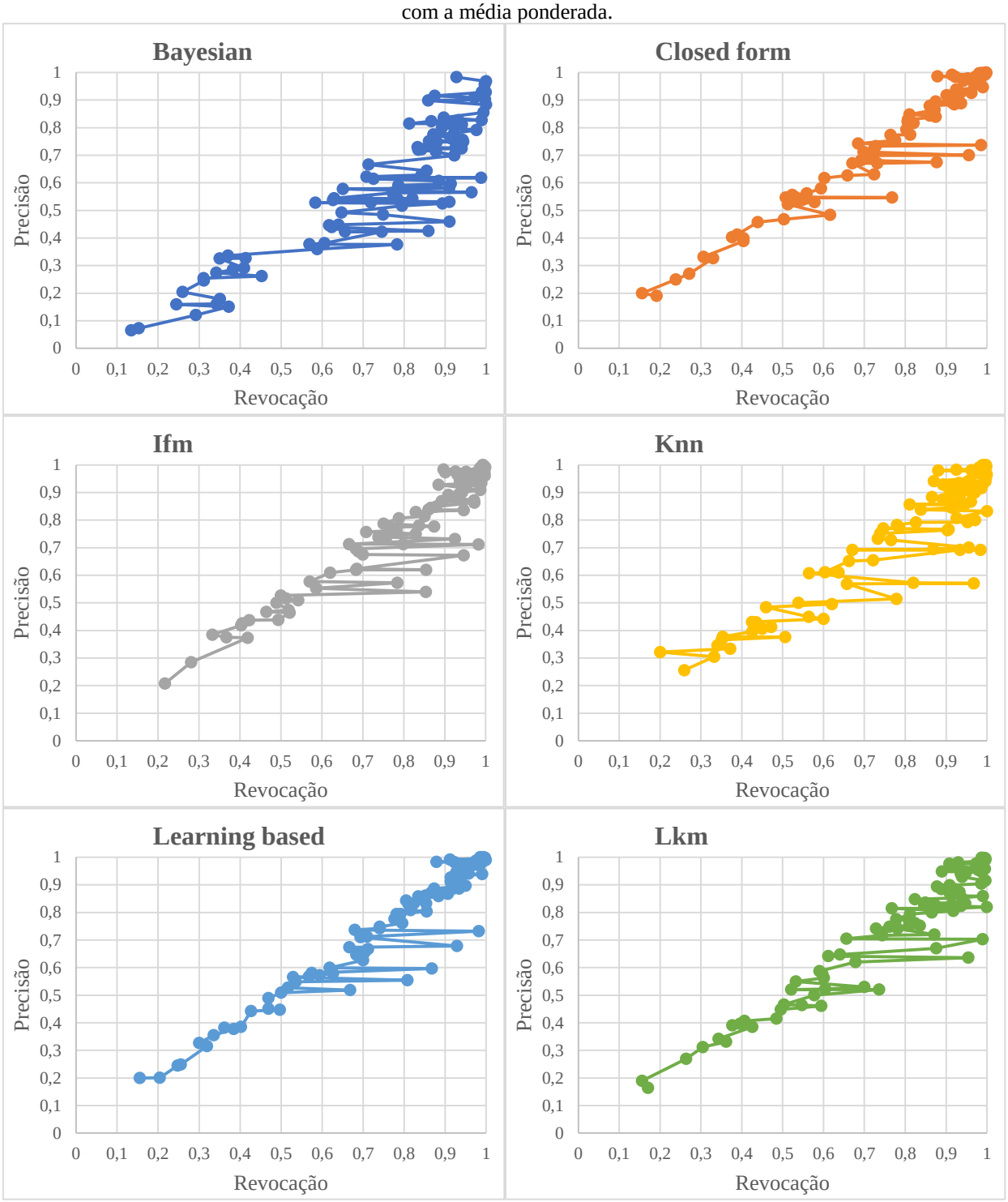
Todavia, conforme descrito anteriormente trata-se de uma avaliação global, por este motivo os resultados individuais de precisão e revocação para o conjunto de imagens avaliadas são detalhados nos dos gráficos de distribuição da Figura 53 e Figura 54. Ao confrontar os gráficos, percebe-se que o algoritmo *Bayesian* apresenta um comportamento atípico no crescimento da curva comparado aos demais, por este motivo algumas das discussões expostas a seguir podem desconsidera-lo durante a análise dos resultados.

Figura 53: Gráfico de curvas para precisão e revocação dos algoritmos *matting* com limiar de corte da binarização 0,5.



Fonte: Compilação do autor.

Figura 54: Gráficos de curvas para precisão e revocação dos algoritmos *matting* com limiar de corte da binarização com a média ponderada.

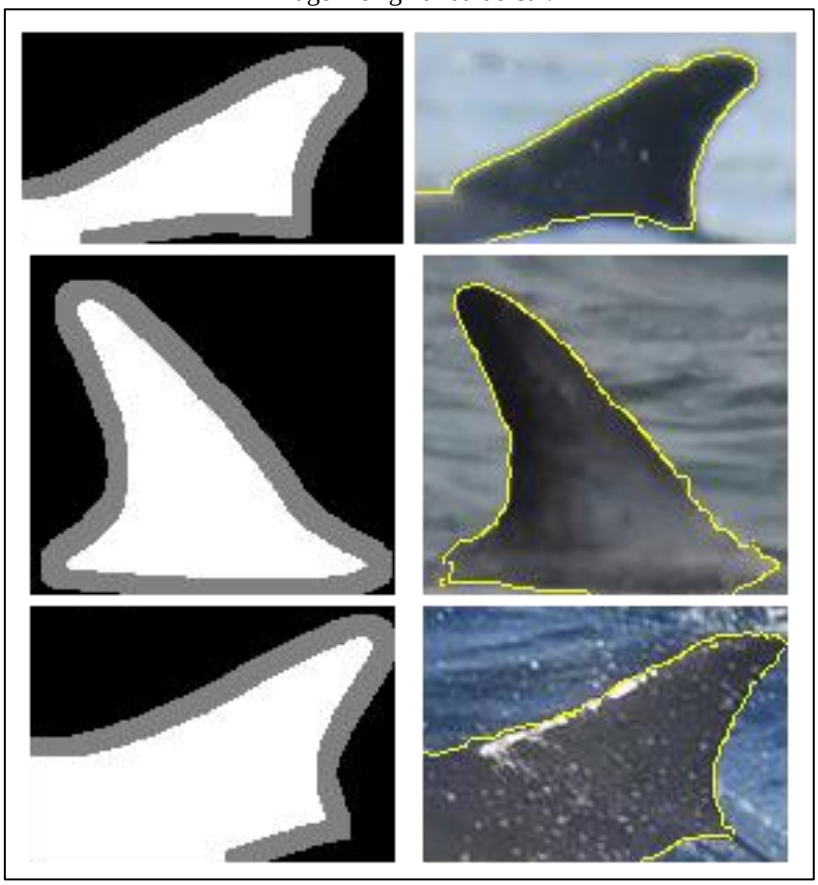


Fonte: Compilação do autor.

Ao analisar as informações dos gráficos, nota-se que o crescimento da linha segue um comportamento compartilhado em todos os resultados. Ou seja, observa-se que valores de precisão andam em paralelo ao crescimento dos valores de revocação. Indicando a presença de um nível de paralelismo na ocorrência de pixels considerados falsos positivos e falsos negativos gerados pelos algoritmos.

Também é perceptível a presença de algumas exceções onde o valor de revocação tende a ser maior que a precisão. Nestes casos, ao analisar visualmente as imagens constatou-se que os algoritmos geraram linhas de contornos excedentes por retratar cenas de indivíduos parcialmente oclusos e com a dorsal próxima a água, característica esta que acarretou na criação de uma área de intersecção do *trimap* entre o corpo do indivíduo e a água, gerando uma região de limites não previstas durante a definição do padrão verdade. Este tipo de situação foi presenciado em 6 imagens do conjunto de dados da avaliação, na Figura 55 são apresentados alguns exemplos da ocorrência.

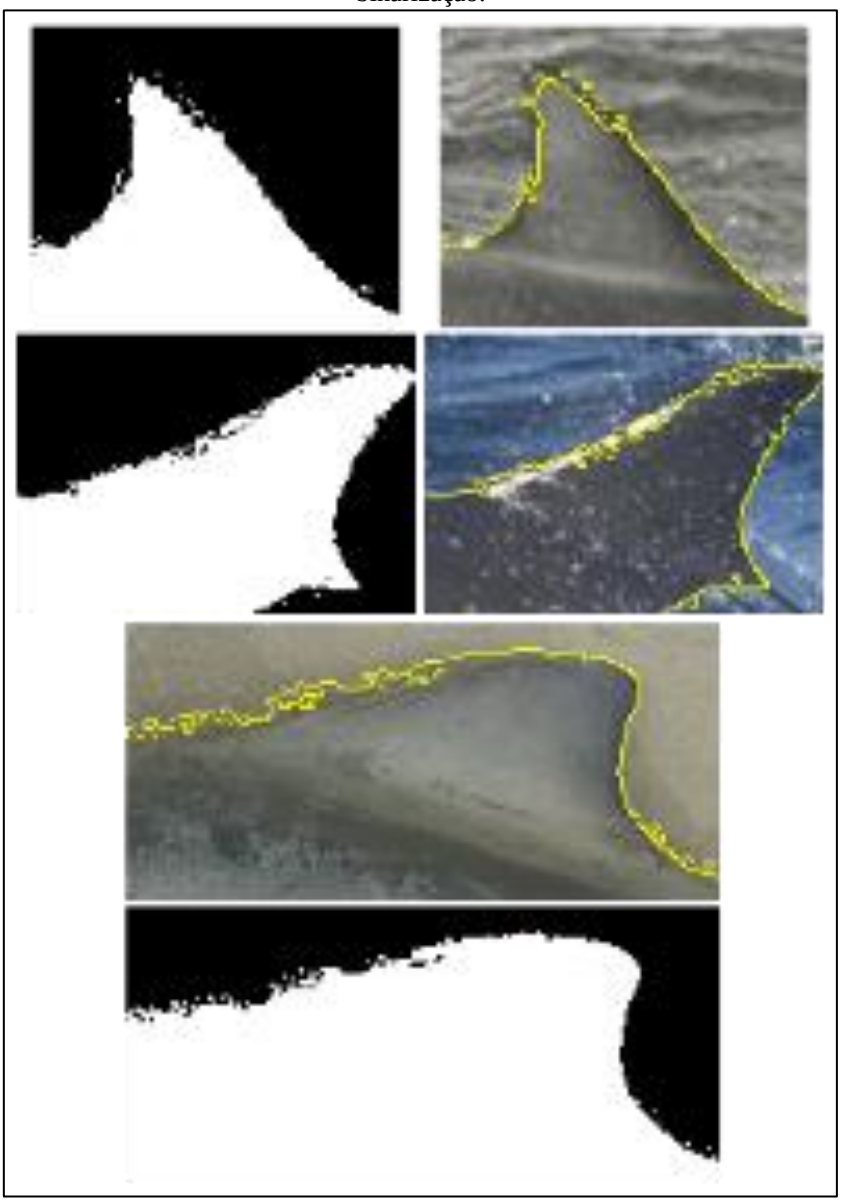
Figura 55: Dorsais com linhas de contornos excedentes, a esquerda o trimap utilizado no processo de *matting* e a direita a linha resultante sobreposta a imagem original da dorsal.



Fonte: Compilação do autor.

Outras 17 imagens apresentaram um comportamento semelhante, porém representam um distanciamento entre a precisão e revocação menor que os citados anteriormente. A verificação destas imagens demostrou que estas divergências estavam ligadas a conjuntos de pixels entre dorsal e o *background* que possuem a mesma intensidade de cores, esta inconsistência criou alguns segmentos indesejados durante a tarefa de binarização (Figura 56). E também foi mais recorrente nos algoritmos knn e lkm conforme pode ser observado nos respectivos gráficos.

Figura 56: Ilhas de segmentos indesejados gerados durante a da tarefa de binarização.



Fonte: Compilação do autor.

A leitura dos gráficos de precisão e revocação também nos levou aos seguintes questionamentos:

1. Do conjunto de dados separados para esta avaliação, quantos resultaram em um valor de precisão abaixo do esperado?
2. Quais fatores influenciaram na ocorrência dos valores de baixa precisão?

Para responder ao primeiro questionamento foi necessário definir um valor de corte que permitisse delimitar quais resultados seriam considerados aceitáveis para esta avaliação. Com intuito de não ser muito rigoroso na escolha deste valor, definiu-se que seriam considerados imprecisos apenas os resultados onde precisão fosse inferior a 0,5. O resultado para este levantamento é apresentado na Tabela 14 e confirma que, com exceção do algoritmo *Bayesian*, os demais casos resultaram em um índice de imprecisão de no máximo 20%. Com destaque ao algoritmo Ifm que combinado ao limiar de corte da binarização usando a média ponderada obteve apenas 14% de resultados imprecisos.

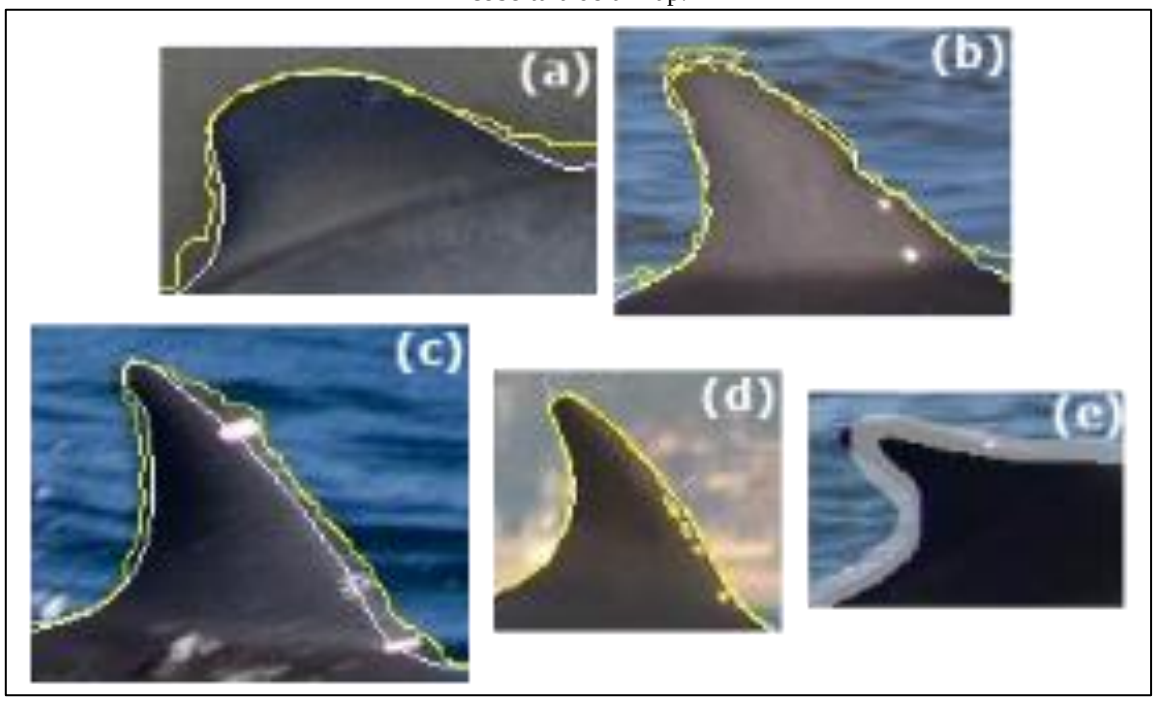
Tabela 14. Levantamento quantitativo das imagens com resultado de precisão (PR) inferior e superior à 0,5.

Algoritmo *matting*	**Limiar de corte binarização**	**PR < 0,5**	**PR >= 0,5**	**% PR < 0,5**	**% PR >= 0,5**
Bayesian	média	30	58	34%	66%
Bayesian	0,5	30	58	34%	66%
Closed form	média	13	75	15%	85%
Closed form	0,5	18	70	20%	80%
Ifm	**média**	**12**	**76**	**14%**	**86%**
Ifm	0,5	15	73	17%	83%
Knn	média	17	71	19%	81%
Knn	0,5	17	71	19%	81%
Learning based	média	14	74	16%	84%
Learning based	0,5	17	71	19%	81%
Lkm	média	15	73	17%	83%
Lkm	0,5	15	73	17%	83%

Os valores descritos Tabela 14 referem-se a um conjunto de 32 imagens que resultaram no índice de precisão inferior a 0,5, isso corresponde a 36% da base de dados construída para esta avaliação. Ao efetuar uma verificação visual nas imagens para descobrir os fatores que

influenciaram no baixo desempenho, apurou-se que 66% destas apresentaram algum tipo de deslocamento na linha de contorno resultante dos algoritmos *matting*, devido a correlação de alguns conjuntos de pixels que compartilhavam de tonalidades de cores aproximadas entre o *background* e a dorsal (Figura 57a e Figura 57b). Também se constatou que em 22% dos casos a imprecisão estava ligada a problemas de foco ou reflexo da luz (Figura 57c e Figura 57d), os 12% restantes correspondem a pequenas inconsistências relacionadas a área de cobertura do *trimap* (Figura 57e).

Figura 57: Imagens com resultado de precisão inferior a 0,5, as linhas brancas correspondem ao padrão verdade e as amarelas os resultados da extração da linha de contorno. (a) e (b) pixels com pouco contraste foreground e background; (c) imagem tremida; (d) interferência do reflexo da luz; (e) erro na área de cobertura do trimap.



Fonte: Compilação do autor.

As análises dos resultados desta seção apontaram que quase todos os algoritmos *matting* utilizados durante a avaliação apresentaram um bom desempenho. Bem como demostrou através do levantamento quantitativo dos resultados imprecisos que a configuração de limiar de corte da binarização com a média ponderada tende a reduzir a imprecisão gerada em alguns algoritmos. A avaliação também revelou que a principal causa de inconsistências geradas pelo processo de *matting* estão ligadas ao entrelaçamento dos pixels do *foreground* e *background*, devido ao baixo contraste entre as regiões.

6 CONCLUSÃO

Pode-se observar no decorrer do desenvolvimento do presente trabalho, que os conceitos da metodologia proposta por Hughes e Burghardt (2016) também se aplicam ao problema de pesquisa desta dissertação. Contudo, para atender o principal objetivo deste trabalho, algumas das etapas de construção do processo automatizado de extração das características de identificação da nadadeira dorsal, estes conceitos foram abordados por outra perspectiva.

Ou seja, o uso de técnicas sofisticadas de visão computacional que se beneficiam da versatilidade dos modelos de redes neurais artificias, bem como a adoção de ferramentas que facilitaram no desenvolvimento de uma solução para o problema proposto. Proporcionou a construção de um processo automatizado para a etapa de extração das características de identificação das nadadeiras dorsais cetáceos, através do uso de técnicas de aprendizado de máquina consideradas no atual momento como estado da arte para este campo de pesquisa.

Entre as diferentes técnicas de visão computacional adotadas durante a construção do processo automatizado, duas se destacam por serem implementações que não foram abordadas nos trabalhos relacionados, bem como permitiu atender os critérios do primeiro objetivo específico. Trata-se das técnicas de detecção de objetos e segmentação semântica, que foram incorporadas ao trabalho através da criação de modelos treinados a partir de redes neurais convolucionais, com um corpus específico de imagens digitais de cetáceos. Esta abordagem possibilitou implementar um novo método para detectar e extrair as dorsais de imagens digitais, além de fornecer o material necessário para a avaliação do método proposto.

Conforme fora apresentado durante a análise dos resultados, todos os modelos de redes neurais utilizados na detecção de objetos obtiveram bons resultados, principalmente na detecção da dorsal onde a diferença na média de precisão entre os modelos foi pequena. Porém na avaliação global o modelo SSD se destacou aos demais, por este motivo foi o escolhido para integrar a versão final da ferramenta proposta. Também é importante salientar que ao explorar a detecção de outras classes de objetos, pode-se observar o potencial uso deste recurso, no desenvolvimento de uma ferramenta de busca de cetáceos em imagens digitais envolvendo grandes bases de dados ambientais.

No que diz respeito à etapa de segmentação o modelo atendeu as expectativas deste trabalho, porém devido a limitação que impossibilita separar objetos sobrepostos, faz-se necessário explorar a implementação de outros modelos, como por exemplo, a segmentação de instancias.

Para atender ao segundo objetivo específico, incorporou-se a técnica de refinamento de linha de contorno adotada por Hughes e Burghardt (2016). Portanto, seis algoritmos *matting* descritos na literatura foram incluídos na etapa de extração da linha de contorno da dorsal. Em uma comparação rígida dos resultados obtidos para esta etapa, pode-se dizer que o algoritmo *Ifm* superou os demais, bem como apresentou um desempenho melhor que o algoritmo *Learning Based* adotado no trabalho de Hughes e Burghardt (2016). Contudo, não se pode afirmar que este seria o melhor algoritmo para o problema proposto, dado a ocorrência de resultados equilibrados obtidos durante a avaliação, onde dos seis algoritmos avaliados apenas o *Bayesian* não se aplica ao contexto do problema.

Esta pequena diferença apresentada pelos resultados, pode estar relacionada ao fato de que a avaliação foi realizada com um conjunto pequeno de dados de teste. Portanto recomenda-se que em trabalhos futuros a avaliação seja efetuada em um conjunto maior de dados. Por outro lado, a análise individual dos resultados para o mesmo conjunto de dados avaliados, demonstrou que independente das condições ambientais retratadas nas cenas, os algoritmos são ineficientes quando aplicados a imagens de baixa qualidade ou com pouco contraste entre o *foreground* e *background*.

O terceiro e último objetivo específico consiste no desenvolvimento de uma ferramenta que disponibilize os dados da extração das linhas de contornos das dorsais para que possam ser utilizadas por qualquer software que seja capaz de utilizar esta informação na etapa de identificação individual. Este objetivo foi atendido ao longo da construção de cada etapa do processo de automatização do extrator de características de identificação individual dos cetáceos, e pode ser incorporado a qualquer método de identificação individual descrito nos trabalhos relacionados.

Em relação as perguntas de pesquisa, esta dissertação demonstrou que foi possível implementar uma solução similar ao método proposto por Hughes e Burghardt (2016), em uma ferramenta de extração das características de identificação individual de cetáceos. Bem como permitiu evidenciar a eficiência das técnicas de visão computacional empregadas a imagens com condições ambientais adversas, através da avaliação quantitativa dos resultados obtidos em cada etapa do processo.

6.1 CONTRIBUIÇÕES

A principal contribuição que este trabalho trouxe para a área de computação foi o desenvolvimento de um processo automatizado para extração das características de identificação das nadadeiras dorsais para cetáceos, utilizando técnicas de visão computacional que misturam algoritmos clássicos e de aprendizado de máquina para produção de recursos computacionais considerados como estado da arte pelos pesquisadores da área.

Este trabalho também contribuiu indiretamente com a possibilidade de criação de novas ferramentas para gestão ambiental em trabalhos futuros. Ou seja, ao explorar a detecção de múltiplos objetos, observou-se o potencial desta técnica na produção de ferramentas que auxiliem no monitoramento ambiental de cetáceos. Um exemplo disto seria a busca e identificação de cetáceos em grandes bases de dados de imagens ambientais, e monitoramento de cetáceos em tempo real utilizando imagens provenientes de câmeras de vídeo instaladas em regiões costeiras, portos, etc.

Outra contribuição deixada foi a criação de um corpus de imagens de cetáceos, baseado em repositórios de dados ambientais de extrema importância no que diz respeito ao monitoramento ambiental e controle populacional. Contendo as anotações de caixa delimitadora para as quatro classes descritas no desenvolvimento, além de um número considerável de segmentos dos indivíduos criados manualmente e que podem ser reaproveitados em trabalhos futuros.

6.2 SUGESTÕES PARA TRABALHOS FUTUROS

Ao finalizar este trabalho obteve-se o conhecimento de que algumas etapas do processo automatizado de extração das características de identificação das nadadeiras dorsais para animais da ordem dos cetáceos, poderiam ser melhorados ao efetuar alguns ajustes.

Sendo estes:

- Melhorar o desempenho dos modelos de detecção de objetos e segmentação, adicionado um número maior de exemplos durante o processo de treinamento;
- Implementar um modelo de segmentação de instâncias para atender a limitação deixada pelo modelo de segmentação semântica;

- Incluir novas classes de objetos presenciados nas cenas ambientadas pelas imagens, visando reduzir os erros de detecção de objetos e as inconsistências geradas pela etapa de segmentação; e
- Ampliar o conjunto de dados de teste e validação dos algoritmos empregados na etapa de extração da linha de contorno da dorsal.

Para atender as recomendações listadas, sugere-se a ampliação do corpus de imagens de cetáceos efetuando uma nova consulta nas bases de dados citadas neste trabalho, ou buscando novas fontes de dados ambientais.

REFERÊNCIAS

AKSOY, Y.; AYDIN, T. O.; POLLEFEYS, M. Designing Effective Inter-Pixel Information Flow for Natural Image Matting. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, Honolulu. **Proceedings…** Honolulu: Hawaii Convention Center, 2017. p. 228-236.

ANDREOTTI, S. et al. Semi-automated software for dorsal fin photographic identification of marine species: application to Carcharodon carcharias. **Marine Biodiversity**, 2017. Disponível em: <https://link.springer.com/article/10.1007/s12526-017-0634-2>. Acesso em: 25 de jul. 2018.

ARAABI, B. N. et al. A String Matching Computer-Assisted System for Dolphin Photoidentification. **Annals of Biomedical Engineering**. [S.I.], v. 28, n. 1, p. 1269-1279, ago. 2000.

ARBELAEZ, P. et al. Contour Detection and Hierarchical Image Segmentation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.I.], v. 33, n. 5, p. 898-916, ago. 2010.

ARBELÁEZ, P. et al. Multiscale Combinatorial Grouping. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014., 2014, Columbus. **Proceedings…** Columbus: USA, 2014. p. 328-335.

ARZOUMANIAN Z.; HOLMBERG, J.; NORMAN, B. An astronomical pattern-matching algorithm for computer-aided identification of whale sharks *Rhincodon typus*. **Journal of Applied Ecology**. [S.I.], v. 42. N. 1, p. 999-1011, 2005.

BARRETO, A. S. et al. Using GIS To Manage Cetacean Strandings. **Journal of Coastal Research**. [S.I.], v. SI 39, p. 1643-1645, 2006.

BEARZI, E. et al. Occurrence and present status of coastal dolphins (*Delphinus delphis* and *Tursiops truncatus*) in the eastern Ionian Sea. **Aquatic Conservation: Marine and Freshwater Ecosystems**. [S.I.], v. 15, n. 3, p. 243-257, mai. 2005.

BODA, J.; PANDYA, D. A Survey on Image Matting Techniques. In: 2018 International Conference on Communication and Signal Processing (ICCSP), 2018, Chennai. **Proceedings…** Chennai: India, 2018. p. 765-770.

BREIMAN, L. Random Forests. **Machine Learning**. [S.I.], v. 45, n. 1, p. 5-32, out. 2001.

CANNY, J. A Computational Approach to Edge Detection. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.I.], v. 8, n. 6, p. 679-698, nov. 1986.

CARTER, S. J. B. et al. Automated marine turtle photograph identification using artificial neural networks, with application to green turtles. **Journal of Experimental Marine Biology and Ecology**. [S.I.], v. 452, n. 1, p. 105-110, mar. 2014.

CARVAJAL-GÁMEZ, B. E. et al. Photo-id of blue whale by means of the dorsal fin using clustering algorithms and color local complexity estimation for mobile devices. **EURASIP Journal on Image and Video Processing**. [S.I.], v. 2017, n. 1, Não paginado, dez. 2017.

CARVAJAL-GAMEZ, B. E.; GALLEGOS-FUNES, F. J.; ROSALES-SILVA, A. J. Color local complexity estimation based steganographic (CLCES) method**. Expert Systems with Applications**. [S.I.], v. 40, n. 4, p. 1132-1142, mar. 2013.

CHEN, L. et al. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. **arXiv**. Mai. 2017. Disponível em: <https://arxiv.org/abs/1606.00915>. Acesso em: 27 mai. 2019.

CHEN, L. et al. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: Computer Vision – ECCV, 15., 2018, Munich. **Proceedings…** Munich: Germany, 2018. p. 833-851.

CHEN, Q.; LI, D.; TANG, C. KNN Matting. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.I.], v. 35, n. 9, p. 2175-2188, jan. 2013.

CHUANG, Y. et al. A Bayesian approach to digital matting. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), [S.I.], 2001, Kauai. **Proceedings…** Kauai: USA, 2001. p. 264-271.

COCO: Detection evaluation. In: COCO, 2015. Disponível em: <http://cocodataset.org/#detection-eval>. Acesso em: 12 jul. 2019.

CORDTS, M. et al. The Cityscapes Dataset. In: CVPR Workshop on The Future of Datasets in Vision, [S.I.], 2015. Boston: Estados Unidos, 2015. Disponível em: <https://www.cityscapes-dataset.com/wordpress/wp-content/papercite-data/pdf/cordts2015cvprw.pdf>. Acesso em: 12 jun. 2019.

DAI, J. et al. R-FCN: object detection via region-based fully convolutional networks. In: Proceeding NIPS'16 Proceedings of the 30th International Conference on Neural Information Processing Systems, 30., 2016, Barcelona. **Proceedings…** Barcelona: Espanha, 2016. p. 379-387.

DARWIN. In: Eckerd College: Department of Computer Science, 1993. Disponível em: <http://darwin.eckerd.edu>. Acesso em: 05 fev. 2018.

DeepLab: Deep Labelling for Semantic Image Segmentation. In: GitHub, 2018. Disponível em: <https://github.com/tensorflow/models/tree/master/research/deeplab>. Acesso em: 27 mai. 2019.

DUTTA, A.; GUPTA, A.; ZISSERMAN, A. VGG Image Annotator VIA. VIA. Set. 2016. Disponível em: <http://www.robots.ox.ac.uk/~vgg/software/via/>. Acesso em: 12 jun. 2019.

EVERINGHAM, M. et al. The Pascal Visual Object Classes (VOC) Challenge. **International Journal of Computer Vision**. [S.I.], v. 88, n. 2, p. 303-338, jun. 2010.

FRIDAY, N. et al. Measurement of Photographic Quality and Individual Distinctiveness for the Photographic Identification of Humpback Whales, *Megaptera novaeangliae*. **Marine Mammal Science**. [S.I.], v. 16, n. 2, p. 355-374, abr. 2000.

GARCIA-GASULLA, D. et al. An Out-of-the-box Full-network Embedding for Convolutional Neural Networks. **arXiv**. Mai. 2017. Disponível em: <https://arxiv.org/abs/1705.07706>. Acesso em: 27 mai. 2019.

GARCIA-GASULLA, D. et al. An Out-of-the-box Full-network Embedding for Convolutional Neural Networks. **arXiv**. Mai. 2017. Disponível em: <https://arxiv.org/abs/1705.07706>. Acesso em: 27 mai. 2019.

GENOV, T. et al. Novel method for identifying individual cetaceans using facial features and symmetry: A test case using dolphins. **Marine Mammal Science**. [S.I], v. 34, n. 2, p. 514-528, abr. 2018.

GILMAN, A. et al. Computer-assisted Recognition Of Dolphin Individuals Using Dorsal Fin Pigmentations. In: International Conference on Image and Vision Computing New Zealand (IVCNZ), 2016, Palmerston. **Proceedings…** Palmerston North: Massey University Palmerston North Campus, 2016. p. 255-260.

GIRSHICK, R. et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In: 2014 IEEE International Conference on Computer Vision Workshops, 2014, Columbus. **Proceedings…** Greater Columbus Convention Center: USA, 2014. p. 580-587.

GIRSHICK, R. Fast R-CNN. **arXiv**. Abr. 2015. Disponível em: <https://arxiv.org/abs/1504.08083>. Acesso em: 25 jul. 2019.

GUO, Y. et al. A review of semantic segmentation using deep neural networks. **International Journal of Multimedia Information Retrieval**. [S.I.], v. 7, n. 2, p. 87–93, jun. 2018.

HALE, S. **Unsupervised Thresholding for Automatic Extraction of Dolphin Dorsal Fin Outlines from Digital Photographs in DARWIN**. 2008. 48 f. Trabalho de conclusão de curso (Graduação) - Bacharelado em Ciência da Computação, Eckerd College St. Petersburg, Florida, 2008.

HALLORAN, K. M.; MURDOCH, J. D.; BECKER, M. S. Applying computer-aided photo-identification to messy datasets: a case study of Thornicroft's giraffe (Giraffa camelopardalis thornicrofti). **African Journal of Ecology**. [S.I.], v. 53, n. 2, p. 147-155, jul. 2014.

HARIHARAN, B. et al. Simultaneous Detection and Segmentation. In: Computer Vision – ECCV, 13., 2014, Zurich. **Proceedings…** Zurich: Switzerland, 2014. p. 297-312.

HE, K. et al. Deep Residual Learning for Image Recognition. **arXiv**. Des. 2015. Disponível em: <https://arxiv.org/abs/1512.03385>. Acesso em: 30 mai. 2019.

HE, K.; SUN, J.; TANG, X. Fast matting using large kernel matting Laplacian matrices. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, San Francisco. **Proceedings…** San Francisco: 5 Embarcadero Center, 2010. p. 2165-2272.

HEIDE-JØRGENSEN, M. P. et al. Long-term tag retention on two species of small cetaceans. **Marine Mammal Science**. [S.I.], v. 33, n. 3, p. 713-725, jul. 2017.

HILLMAN, G. R. et al. "Finscan", a Computer System for Photographic Identification of Marine Animals. In: Proceedings of the SecondJoint EMBSBMES Conference, 2002, Houston. **Proceedings…** Houston: USA, 2002. p. 1065-1066.

HOOVER. A. L. et al. Comparing Acoustic Tag Attachments Designed for Mobile Tracking of Hatchling Sea Turtles. **Frontiers in Marine Science**. [S.I.], v. 4, jul. 2017. Disponível em: <https://www.frontiersin.org/articles/10.3389/fmars.2017.00225/full>. Acesso em: 25 abr. 2018.

HOWARD, A. G. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. **arXiv**. Abr. 2017. Disponível em: <https://arxiv.org/abs/1704.04861>. Acesso em: 30 mai. 2019.

HUANG, J. et al. Speed/accuracy trade-offs for modern convolutional object detectors. **arXiv**. Nov. 2016. Disponível em: <https://arxiv.org/abs/1611.10012>. Acesso em: 27 mai. 2019.

HUGHES, B.; BURGHARDT, T. Affinity Matting for Pixel-accurate Fin Shape Recovery from Great White Shark Imagery. In: Proceedings of the Machine Vision of Animals and their Behaviour (MVAB), 2015, Swansea. **Proceedings…** Swansea: United Kingdom, 2015. p. 8.1-8.8.

HUGHES, B.; BURGHARDT, T. Automated Visual Fin Identification of Individual Great White Sharks. **International Journal of Computer Vision**. [S.I.], v. 122, n. 3, p. 542-557, out. 2016.

INATURALIST: iNaturalist. In: iNaturalist, 2018. Disponível em: <https://www.inaturalist.org/>. Acesso em: 13 jul. 2018.

IRVINE, A. B.; WELLS, R. S.; SCOTT, M. D. An Evaluation of Techniques for Tagging Small Odontocete Cetaceans. **Fishery Bulletin**. [S.I.], v. 80, n. 1, p. 135-143, 1982.

KELLY, M. J. Computer-Aided Photograph Matching in Studies Using Individual Identification: An Example from Serengeti Cheetahs. **Journal of Mammalogy**. [S.I.], v. 82, n. 2, p. 440-449, mai. 2001.

KREHO, A. et al. Computer assisted feature extraction for dolphin identification. In: Proceedings of the International Conference on Imaging Science, Systems, and Technology, 1997, Las Vegas. **Proceedings…** Las Vegas: USA, 1997. p. 440-445.

KUMAR, S. et al. **Animal Biometrics:** Techniques and Applications. 1. ed. Singapore: Springer Nature Singapore Pte Ltd, 2017.

LabelImg. In: GitHub, 2015. Disponível em: <https://github.com/tzutalin/labelImg>. Acesso em: 6 ago. 2019.

LAHIRI, M. et al. Biometric Animal Databases from Field Photographs: Identification of Individual Zebra in the Wild. In: ACM International Conference on Multimedia Retrieval (ICMR), 1., Trento, 2011. **Proceedings…** Trento: University of Trento, 2011. Não paginado.

LATEEF, F; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. **Neurocomputing**. [S.I.], v. 338, n. 1, p. 321-348, abr. 2019.

LEVIN, A.; LISCHINSKI, D.; WEISS, Y. A Closed-Form Solution to Natural Image Matting. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.I.], v. 30, n. 2, p. 228-242, dez. 2007.

LIN, T. et al. Microsoft COCO: Common Objects in Context. In: Computer Vision – ECCV 2014, 2014, Zurich. **Proceedings…** Zurich: Switzerland, 2014. p. 740-755.

LIU, W. et al. SSD: Single Shot MultiBox Detector. In: Computer Vision – ECCV 2016, 2016, Amsterdam. **Proceedings…** Amsterdam: Netherlands, 2016. p. 21-37.

LOFFE, S.; SZEGEDY, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. **arXiv**. Fev. 2015. Disponível em: <https://arxiv.org/abs/1502.03167>. Acesso em: 30 mai. 2019.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, Boston. **Proceedings…** Boston: Hynes Convention Center, 2015. p. 3431-3440.

LOWE, D. G.; MCCANN, S. Local Naive Bayes Nearest Neighbor for image classification. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, Providence. Providence: USA, 2012. p. 3650-3656.

MARKOWI, T. M.; HARLIN, A. D.; WURSIG, B. Digital Photography Improves Efficiency of Individual Dolphin Identification. **Marine Mammal Science**. [S.I.], v. 19, n. 1, p. 217-223, jan. 2003.

MARTIN, D.; FOWLKES, C.; MALIK, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. **IEEE Transactions on Pattern Analysis and Machine Intelligence**. [S.I.], v. 26, n. 11, p. 530-549, mai. 2004.

NIELSEN, M. A. **Neural Networks and Deep Learning**. 1. ed. Determination Press, 2015. Disponível em: <http://neuralnetworksanddeeplearning.com/index.html> Acesso em: 6 ago. 2019.

NORMAN, S. A. et al. Assessment of wound healing of tagged gray (*Eschrichtius robustus*) and blue (*Balaenoptera musculus*) whales in the eastern North Pacific using long- term series of photographs. **Marine Mammal Science**. [S.I.], v. 34, n. 1, p. 27-53, jan. 2018.

OSBOURN, M. S. et al. Use of Fluorescent Visible Implant Alphanumeric Tags to Individually Mark Juvenile Ambystomatidae Salamanders. **Herpetological Review**. [S.I.], v. 42, n. 1, p. 43-47, mar. 2011.

PASCAL VOC. In: VOC2012, 2012. Disponível em: <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/>. Acesso em: 12 jun. 2019.

PERRIN, W.; WÜRSIG, B.; THEWISSEN J.G.M. **Encyclopedia of Marine Mammals**. 2. ed. Cambridge: Academic Press, 2008.

PLATANIOTIS, K. N.; VENETSANOPOULOS, A. N. **Color Image Processing and Applications**. 1. ed. Berlin: Springer-Verlag, 2000.

PMP-BS: Projeto de Monitoramento de Praias da Bacia de Santos. In: PMP-BS, 2017. Disponível em: < http://pmp.acad.univali.br>. Acesso em: 17 dez. 2018.

POLLICELLI, D.; COSCARELLA, M.; DELRIEUX, C. Wild Cetacea Identification using Image Metadata. **Journal of Computer Science and Technology (JCS&T)**. [S.I.], v. 17, n. 1, p. 79-84, abr. 2017.

REN, S. et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceeding NIPS'15 Proceedings of the 28th International Conference on Neural Information Processing Systems, 28., 2015, Montreal. **Proceedings…** Montreal: Montreal Convention Center, 2015. p. 91-99.

ROTHER, C.; KOLMOGOROV, V.; BLAKE, A. Grabcut: Interactive foreground extraction using iterated graph cuts. **ACM Transactions on Graphics (TOG)**. [S.I.], v. 23, n. 3, p. 309-314, ag. 2004.

SARAUX, C. et al. Reliability of flipper-banded penguins as indicators of climate change. **Nature**. Jan. 2011. Disponível em: <https://www.nature.com/articles/nature09630>. Acesso em: 5 mai. 2018.

Shanmugamani, R. **Deep Learning for Computer Vision**. 1. ed. Birmingham: Packt Publishing, 2018.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. **arXiv**. Set. 2014. Disponível em: <https://arxiv.org/abs/1409.1556>. Acesso em: 30 mai. 2019.

SINGH, S.; JALAL, A. S. Digital Image Matting: A Review. **International Journal of Computer Vision and Image Processing**. [S.I.], v. 3, n. 4, p. 16-36, out. 2013.

SISPMC: PMC-BS: Projeto de Monitoramento de Cetáceos na Bacia de Santos. In: SISPMC, 2016. Disponível em: <http://sispmc.socioambiental.com.br/sispmc/>. Acesso em: 10 out. 2017.

SPEED, C. W.; MEEKAN, M. G.; BRADSHAW, C. J. A. Spot the match – wildlife photo-identification using information theory. **Frontiers in Zoology**. Jan. 2007. Disponível em <https://frontiersinzoology.biomedcentral.com/articles/10.1186/1742-9994-4-2>. Acesso em: 11 jun. 2018.

SZEGEDY, C. et al. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. **arXiv**. Fev. 2016. Disponível em: <https://arxiv.org/abs/1602.07261>. Acesso em: 30 mai. 2019.

SZEGEDY, C. et al. Rethinking the Inception Architecture for Computer Vision. **arXiv**. Des. 2015. Disponível em: <https://arxiv.org/abs/1512.00567>. Acesso em: 30 mai. 2019.

Tensorflow Object Detection API. In: GitHub, 2017. Disponível em: <https://github.com/tensorflow/models/tree/master/research/object_detection>. Acesso em: 30 nov. 2018.

TITOVA, O. V. et al. Photo-identification matches of humpback whales (Megaptera novaeangliae) from feeding areas in Russian Far East seas and breeding grounds in the North Pacific. **Marine Mammal Science**. [S.I.], v. 34, n. 1, p. 100-112, jan. 2018.

UIJLINGS, J. R. R. et al. Selective Search for Object Recognition. **International Journal of Computer Vision**. [S.I.], v. 104, n. 2, p. 154-171, set. 2013.

WANG, J.; COHEN, M. F. Image and video matting: a survey. **Journal Foundations and Trends® in Computer Graphics and Vision**. [S.I.], v. 3, n. 2, p. 97-175, jan. 2007.

WEIDEMAN, H. J. et al. Integral Curvature Representation and Matching Algorithms for Identification of Dolphins and Whales. In: 2017 IEEE International Conference on Computer Vision Workshops, 2017, Venice. **Proceedings…** Venice: Palazzo del Cinema - Venice Convention Center, 2017. p. 2831-2839.

Wildbook for Whale Sharks. In: WILDBOOK, 2018. Disponível em: <https://www.whaleshark.org/>. Acesso em: 13 jul. 2018.

Wildbook: Software to Combat Extinction. In: WILDME, 2016. Disponível em: <http://www.wildbook.org/doku.php>. Acesso em: 27 abr. 2018.

YOSINSKI, J. et al. How transferable are features in deep neural networks?. In: Proceeding NIPS'14 Proceedings of the 27th International Conference on Neural Information Processing Systems, 27., 2014, Montreal. **Proceedings…** Montreal: Montreal Convention Center, 2014. p. 3320-3328.

YURKOV, A. O.; CHERNUKHA, I. V. Automated Identification and Recognition of Right Whales. **TAAC**. Nov. 2015. Disponível em: <https://taac.org.ua/files/a2015/proceedings/UA-1-Ivan%20Chernukha-506.pdf>. Acesso em: 7 fev. 2018.

ZHANG, X. et al. Robust image corner detection based on scale evolution difference of planar curves. **Pattern Recognition Letters**. [S.I.], v.30, n. 4, p. 449-455, mar. 2009.

ZHAO, Z. et al. Object Detection With Deep Learning: A Review. **arXiv**. Jul. 2018. Disponível em: <https://arxiv.org/abs/1807.05511>. Acesso em: 24 jul. 2019.

ZHELEZNIAKOV, A. et al. Segmentation of Saimaa Ringed Seals for Identification Purposes. In: International Symposium on Visual Computing, 11., 2015, Las Vegas. **Proceedings…** Las Vegas: Monte Carlo Resort & Casino, 2015. p. 227-236.

ZHENG, Y.; KAMBHAMETTU, C. Learning based digital matting. In: IEEE International Conference on Computer Vision, 12., 2009, Kyoto. **Proceedings…** Kyoto: Japão, 2009. p. 889-896.

GLOSSÁRIO

Alpha	Alfa é um valor que define a opacidade de um pixel numa imagem.
API	API é um conjunto de rotinas e padrões de programação para acesso a um aplicativo de software ou plataforma. A sigla API refere-se ao termo em inglês "Application Programming Interface" que significa em tradução para o português "Interface de Programação de Aplicativos".
Background	Termo em inglês utilizado para descrever a região imagem ou contexto da cena não ocupado pelo foreground.
Binarização	Classificação dos pixels de uma imagem em apenas duas cores, como por exemplo, preto e branco.
Camada alpha	Conjunto de valores que definem as intensidades de opacidades para os pixels de uma imagem.
CMYK	Abreviatura do sistema de cores subtrativas formado por Ciano (Cyan), Magenta (Magenta), Amarelo (Yellow) e Preto (Black (Key ou para não confusão com o B de "Blue" no padrão Hi-Fi com RGB)).
Corpus	Um conjunto de documentos ou dados sobre determinado assunto.
Época	Unidade de medida que descreve a quantidade de vezes que todos os dados de treinamento de uma rede neural artificial foram totalmente processados.
Foreground	Termo em inglês utilizado para descrever os objetos em primeiro plano de uma imagem ou cena.
Framework	Trata-se de uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um framework pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação.
GB	Unidade de medida de informação que equivale a um bilhão de bytes.
Lote	Unidade de medida que define a quantidade de dados processados em um passo de treinamento de uma rede neural artificial.
Matching	Termo em inglês utilizado para descrever a análise de correspondência entre a imagem analisada e as imagens armazenadas em uma base de dados.
Metadados	Conjunto de dados descritivos das características físicas ou morfológicas de um grupo de indivíduos.
Passo de treinamento	Termo que define a execução de um determinado algoritmo uma ou mais vezes durante a atividade de treinamento de uma rede neural artificial.

Perseptrom	O perceptron é um classificador binário análogo a um neurónio, que mapeia sua entrada *x* (um vetor de valor real) para um valor de saída *f(x)* (um valor binário simples) através de uma matriz.
Peso	Representação análoga das sinapses neurais em uma rede neural artificial.
Pixel	O menor componente de uma imagem digital.
PNG	Um formato de dados utilizado para imagens que permite comprimi-las sem perda de qualidade e retirar o fundo de imagens com o uso do canal alfa.
Ranking	Termo em inglês que define um processo de posicionamento de itens individuais conforme a sua relevância em uma lista de classificação.
RGB	Abreviatura de um sistema de cores aditivas em que o Vermelho (Red), o Verde (Green) e o Azul (Blue) são combinados de várias formas de modo a reproduzir um largo espectro cromático.
Threshold	Definição utilizada para descrever o limiar de corte de uma função ou tarefa.
top-1	Utilizado para descrever que um determinado elemento foi encontrado corretamente em uma lista de classificação.
top-5	Utilizado para descrever que um determinado elemento foi encontrado entre as cinco posições iniciais de uma lista de classificação.
Viés	Utilizado como sinal de excitação do neurônio em uma rede neural artificial.
XOR	Ou exclusivo ou disjunção exclusiva, conhecido geralmente por XOR ou por EXOR (também XOU ou EOU), é uma operação lógica entre dois operandos que resulta em um valor lógico verdadeiro se e somente se o número de operandos com valor verdadeiro for ímpar.

APÊNDICE A – ARQUIVO DE CONFIGURAÇÃO DO MODELO PRÉ-TREINADO ssd_resnet_50_fpn_coco

```
model {
  ssd {
    num_classes: 4
    image_resizer {
      fixed_shape_resizer {
        height: 640
        width: 640
      }
    }
    feature_extractor {
      type: "ssd_resnet50_v1_fpn"
      depth_multiplier: 1.0
      min_depth: 16
      conv_hyperparams {
        regularizer {
          l2_regularizer {
            weight: 0.000399999989895
          }
        }
        initializer {
          truncated_normal_initializer {
            mean: 0.0
            stddev: 0.0299999993294
          }
        }
        activation: RELU_6
        batch_norm {
          decay: 0.996999979019
          scale: true
          epsilon: 0.0010000000475
        }
      }
      override_base_feature_extractor_hyperparams: true
    }
    box_coder {
      faster_rcnn_box_coder {
        y_scale: 10.0
        x_scale: 10.0
        height_scale: 5.0
        width_scale: 5.0
      }
    }
    matcher {
      argmax_matcher {
        matched_threshold: 0.5
        unmatched_threshold: 0.5
        ignore_thresholds: false
        negatives_lower_than_unmatched: true
        force_match_for_each_row: true
        use_matmul_gather: true
      }
    }
```

```
similarity_calculator {
  iou_similarity {
  }
}
box_predictor {
  weight_shared_convolutional_box_predictor {
    conv_hyperparams {
      regularizer {
        l2_regularizer {
          weight: 0.000399999989895
        }
      }
      initializer {
        random_normal_initializer {
          mean: 0.0
          stddev: 0.00999999977648
        }
      }
      activation: RELU_6
      batch_norm {
        decay: 0.996999979019
        scale: true
        epsilon: 0.0010000000475
      }
    }
    depth: 256
    num_layers_before_predictor: 4
    kernel_size: 3
    class_prediction_bias_init: -4.59999990463
  }
}
anchor_generator {
  multiscale_anchor_generator {
    min_level: 3
    max_level: 7
    anchor_scale: 4.0
    aspect_ratios: 1.0
    aspect_ratios: 2.0
    aspect_ratios: 0.5
    scales_per_octave: 2
  }
}
post_processing {
  batch_non_max_suppression {
    score_threshold: 0.300000011921
    iou_threshold: 0.600000023842
    max_detections_per_class: 100
    max_total_detections: 100
  }
  score_converter: SIGMOID
}
normalize_loss_by_num_matches: true
loss {
  localization_loss {
    weighted_smooth_l1 {
    }
  }
  classification_loss {
```

```
        weighted_sigmoid_focal {
          gamma: 2.0
          alpha: 0.25
        }
      }
      classification_weight: 1.0
      localization_weight: 1.0
    }
    encode_background_as_zeros: true
    normalize_loc_loss_by_codesize: true
    inplace_batchnorm_update: true
    freeze_batchnorm: false
  }
}
train_config {
  batch_size: 32
  data_augmentation_options {
    random_horizontal_flip {
    }
  }
  data_augmentation_options {
    random_crop_image {
      min_object_covered: 0.0
      min_aspect_ratio: 0.75
      max_aspect_ratio: 3.0
      min_area: 0.75
      max_area: 1.0
      overlap_thresh: 0.0
    }
  }
  sync_replicas: true
  optimizer {
    momentum_optimizer {
      learning_rate {
        cosine_decay_learning_rate {
          learning_rate_base: 0.0399999991059
          total_steps: 4250
          warmup_learning_rate: 0.0133330002427
          warmup_steps: 2000
        }

      }
      momentum_optimizer_value: 0.899999976158
    }
    use_moving_average: false
  }
  fine_tune_checkpoint:
"path_to_models/ssd_resnet50_v1_fpn_shared_box_predictor_640x640_coco14_sync_2018_07_03/model.
ckpt"
  from_detection_checkpoint: true
  num_steps: 25000
  startup_delay_steps: 0.0
  replicas_to_aggregate: 8
  max_number_of_boxes: 100
  unpad_groundtruth_tensors: false
}
train_input_reader {
  label_map_path: "path_to_labelmap/labelmap.pbtxt"
```

```
  tf_record_input_reader {
    input_path: "path_to_train_data/train.record-?????-of-00010"
  }
}
eval_config {
  num_examples: 383
  max_evals: 1
  use_moving_averages: false
  metrics_set:"coco_detection_metrics"
  include_metrics_per_category: true
  visualize_groundtruth_boxes: true
  export_path: "path_to_evaluating/result.json"
  keep_image_id_for_visualization_export: true,
  visualization_export_dir: "path_to_visualization/visualization/"
  save_graph: true
  num_visualizations: 383
}
eval_input_reader {
  label_map_path: "path_to_labelmap/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
  tf_record_input_reader {
    input_path: "path_to_eval_data/val.record-?????-of-00005"
  }
}
```

APÊNDICE B – ARQUIVO DE CONFIGURAÇÃO DO MODELO PRÉ-TREINADO rfcn_resnet101_coco

```
model {
  faster_rcnn {
    num_classes: 4
    image_resizer {
      keep_aspect_ratio_resizer {
        min_dimension: 600
        max_dimension: 1024
      }
    }
    feature_extractor {
      type: "faster_rcnn_resnet101"
      first_stage_features_stride: 16
    }
    first_stage_anchor_generator {
      grid_anchor_generator {
        height_stride: 16
        width_stride: 16
        scales: 0.25
        scales: 0.5
        scales: 1.0
        scales: 2.0
        aspect_ratios: 0.5
        aspect_ratios: 1.0
        aspect_ratios: 2.0
      }
    }
    first_stage_box_predictor_conv_hyperparams {
      op: CONV
      regularizer {
        l2_regularizer {
          weight: 0.0
        }
      }
      initializer {
        truncated_normal_initializer {
          stddev: 0.00999999977648
        }
      }
    }
    first_stage_nms_score_threshold: 0.0
    first_stage_nms_iou_threshold: 0.699999988079
    first_stage_max_proposals: 100
    first_stage_localization_loss_weight: 2.0
    first_stage_objectness_loss_weight: 1.0
    second_stage_box_predictor {
      rfcn_box_predictor {
        conv_hyperparams {
          op: CONV
          regularizer {
            l2_regularizer {
              weight: 0.0
```

```
          }
        }
        initializer {
          truncated_normal_initializer {
            stddev: 0.00999999977648
          }
        }
      }
      num_spatial_bins_height: 3
      num_spatial_bins_width: 3
      crop_height: 18
      crop_width: 18
    }
  }
  second_stage_post_processing {
    batch_non_max_suppression {
      score_threshold: 0.300000011921
      iou_threshold: 0.600000023842
      max_detections_per_class: 100
      max_total_detections: 100
    }
    score_converter: SOFTMAX
  }
  second_stage_localization_loss_weight: 2.0
  second_stage_classification_loss_weight: 1.0
 }
}
train_config {
 batch_size: 1
 data_augmentation_options {
  random_horizontal_flip {
  }
 }
 optimizer {
  momentum_optimizer {
   learning_rate {
    manual_step_learning_rate {
     initial_learning_rate: 0.000300000014249
     schedule {
      step: 1
      learning_rate: 0.000300000014249
     }
     schedule {
      step: 900000
      learning_rate: 2.99999992421e-05
     }
     schedule {
      step: 1200000
      learning_rate: 3.00000010611e-06
     }
    }
   }
   momentum_optimizer_value: 0.899999976158
  }
  use_moving_average: false
 }
 gradient_clipping_by_norm: 10.0
 fine_tune_checkpoint: "path_to_models/rfcn_resnet101_coco_2018_01_28/model.ckpt"
```

```
  from_detection_checkpoint: true
  num_steps: 34000
}
train_input_reader {
  label_map_path: "path_to_labelmap/labelmap.pbtxt"
  tf_record_input_reader {
    input_path: "path_to_train_data/train.record-?????-of-00010"
  }
}
eval_config {
  num_examples: 383
  max_evals: 1
  use_moving_averages: false
  metrics_set:"coco_detection_metrics"
  include_metrics_per_category: true
  visualize_groundtruth_boxes: true,
  export_path: "path_to_evaluating/result.json"
  keep_image_id_for_visualization_export: true,
  visualization_export_dir: "path_to_visualization/"
  save_graph: true
  num_visualizations: 383
}
eval_input_reader {
  label_map_path: "path_to_labelmap/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
  tf_record_input_reader {
    input_path: "path_to_eval_data/val.record-?????-of-00005"
  }
}
```

APÊNDICE C – ARQUIVO DE CONFIGURAÇÃO DO MODELO PRÉ-TREINADO faster_rcnn_nas

```
model {
  faster_rcnn {
    num_classes: 4
    image_resizer {
      fixed_shape_resizer {
        height: 768
        width: 1024
      }
    }
    feature_extractor {
      type: "faster_rcnn_nas"
    }
    first_stage_anchor_generator {
      grid_anchor_generator {
        height_stride: 16
        width_stride: 16
        scales: 0.25
        scales: 0.5
        scales: 1.0
        scales: 2.0
        aspect_ratios: 0.5
        aspect_ratios: 1.0
        aspect_ratios: 2.0
      }
    }
    first_stage_box_predictor_conv_hyperparams {
      op: CONV
      regularizer {
        l2_regularizer {
          weight: 0.0
        }
      }
      initializer {
        truncated_normal_initializer {
          stddev: 0.00999999977648
        }
      }
    }
    first_stage_nms_score_threshold: 0.0
    first_stage_nms_iou_threshold: 0.699999988079
    first_stage_max_proposals: 300
    first_stage_localization_loss_weight: 2.0
    first_stage_objectness_loss_weight: 1.0
    initial_crop_size: 17
    maxpool_kernel_size: 1
    maxpool_stride: 1
    second_stage_box_predictor {
      mask_rcnn_box_predictor {
        fc_hyperparams {
          op: FC
          regularizer {
```

```
          l2_regularizer {
            weight: 0.0
          }
        }
        initializer {
          variance_scaling_initializer {
            factor: 1.0
            uniform: true
            mode: FAN_AVG
          }
        }
      }
      use_dropout: false
      dropout_keep_probability: 1.0
    }
  }
  second_stage_post_processing {
    batch_non_max_suppression {
      score_threshold: 0.300000011921
      iou_threshold: 0.600000023842
      max_detections_per_class: 100
      max_total_detections: 100
    }
    score_converter: SOFTMAX
  }
  second_stage_localization_loss_weight: 2.0
  second_stage_classification_loss_weight: 1.0
 }
}
train_config {
 batch_size: 1
 data_augmentation_options {
  random_horizontal_flip {
  }
 }
 optimizer {
  momentum_optimizer {
   learning_rate {
    manual_step_learning_rate {
     initial_learning_rate: 0.000300000014249
     schedule {
      step: 1
      learning_rate: 0.000300000014249
     }
     schedule {
      step: 900000
      learning_rate: 2.99999992421e-05
     }
     schedule {
      step: 1200000
      learning_rate: 3.00000010611e-06
     }
    }
   }
   momentum_optimizer_value: 0.899999976158
  }
  use_moving_average: false
 }
```

```
  gradient_clipping_by_norm: 10.0
  fine_tune_checkpoint: "path_to_models/faster_rcnn_nas_coco_2018_01_28/model.ckpt"
  from_detection_checkpoint: true
  num_steps: 200000
}
train_input_reader {
  label_map_path: "path_to_labelmap/labelmap.pbtxt"
  tf_record_input_reader {
    input_path: "path_to_train_data/train.record-?????-of-00010"
  }
}
eval_config {
  num_examples: 383
  max_evals: 1
  use_moving_averages: false
  metrics_set:"coco_detection_metrics"
  include_metrics_per_category: true
  visualize_groundtruth_boxes: true,
  export_path: "path_to_evaluating/result.json"
  keep_image_id_for_visualization_export: true,
  visualization_export_dir: "path_to_visualization/"
  save_graph: true
  num_visualizations: 383
}
eval_input_reader {
  label_map_path: "path_to_labelmap/labelmap.pbtxt"
  shuffle: false
  num_readers: 1
  tf_record_input_reader {
    input_path: "path_to_eval_data/val.record-?????-of-00005"
  }
}
```

APÊNDICE D – CONFIGURAÇÕES PARA O TREINAMENTO DO DEEPLAB

```
python deeplab/train.py
        --logtostderr
        --training_number_of_steps=50000
        --train_split="train"
        --model_variant="xception_65"
        --atrous_rates=6
        --atrous_rates=12
        --atrous_rates=18
        --output_stride=16
        --decoder_output_stride=4
        --train_crop_size=513
        --train_crop_size=513
        --train_batch_size=1
        --dataset="dolphin"
        --tf_initial_checkpoint="path_to_initial_trained_model/deeplabv3_pascal_train_aug/model.ckpt"
        --train_logdir="path_to_training_log"
        --dataset_dir="path_to_training_dataset"
        --fine_tune_batch_norm=False
```