Pedro Otávio Cardozo de Souza Ribeiro

Comparação de Cenas Subaquáticas a partir Imagens Ácústicas baseada em Aprendizado Profundo

Brasil

2018, Abril

Pedro Otávio Cardozo de Souza Ribeiro

Comparação de Cenas Subaquáticas a partir Imagens Acústicas baseada em Aprendizado Profundo

Universidade Federal do Rio Grande – FURG

Centro de Ciências Computacionais

Mestrado em Engenharia de Computação

Orientador: Prof. Dr. Paulo Lilles Jorge Drews-Jr

Coorientador: Prof[a]. Dr. Silvia Silva da Costa Botelho

Brasil

2018, Abril

Ficha catalográfica

R484c Ribeiro, Pedro Otávio Cardozo de Souza.
Comparação de cenas subaquáticas a partir imagens acústicas baseada em aprendizado profundo / Pedro Otávio Cardozo de Souza Ribeiro. – 2018.
83 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande – FURG, Programa de Pós-Graduação em Computação, Rio Grande/RS, 2018.
Orientador: Dr. Paulo Lilles Jorge Drews Júnior.
Coorientadora: Dra. Silvia Silva da Costa Botelho

1. Aprendizagem profunda 2. Sonares de imageamento frontal 3. Robótica subaquática 4. Redes convolutivas 5. Extração de características 6. Aprendizagem de métrica 7. Correspondência de imagens 8. Recuperação de imagens I. Drews Júnior, Paulo Lilles Jorge II. Botelho, Silvia Silva da Costa III. Título.

CDU 004.896

Catalogação na Fonte: Bibliotecária Vanessa Ceiglinski Nunes CRB 10/2174

FURG MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO RIO GRANDE
CENTRO DE CIÊNCIAS COMPUTACIONAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO
CURSO DE MESTRADO EM ENGENHARIA DE COMPUTAÇÃO

ATA DE SESSÃO DE DEFESA DE DISSERTAÇÃO DE MESTRADO

Ata nº ____/201_

Na data de 20 de abril de 2018, às 15h30min, ocorreu a Sessão de Defesa de Dissertação de Mestrado de Pedro Otávio Cardozo de Souza Ribeiro, que apresentou a dissertação intitulada "Comparação de Cenas Subaquáticas a partir de Imagens Acústicas baseada em Aprendizado Profundo", realizada sob a orientação do Prof. Dr. Paulo Lilles Jorge Drews Junior e coorientação da Profª. Drª. Silvia Silva da Costa Botelho. A banca examinadora foi constituida pelos Profs. Prof. Dr. Douglas Guimarães Macharet (UFMG), Prof. Dr. Rodrigo da Silva Guerra (UFSM), Prof. Dr. Ricardo Nagel Rodrigues (FURG) sob a presidência do orientador. Após a apresentação do trabalho, a banca arguiu o candidato e, a seguir, deliberou pela

(X) aprovação da Dissertação
( ) aprovação da Dissertação, sugerindo modificações no texto
( ) reprovação da Dissertação

Rio Grande, 20 de abril de 2018

Prof. Dr. Douglas Guimarães Macharet

Prof. Dr. Rodrigo da Silva Guerra

Prof. Dr. Ricardo Nagel Rodrigues

Prof. Dr. Paulo Lilles Jorge Drews Junior
Orientador(a)

Profª. Drª. Silvia Silva da Costa Botelho
Coorientador(a)

Agradecimentos

Agradeço primeiramente aos meus pais: a minha mãe, Elen Jane e ao meu pai Luciano, por todo o suporte que recebi durante todos esses anos. É difícil descrever em palavras tamanho sentimento de amor e gratidão.

Agradeço aos meus orientadores, professor Dr. Paulo Lilles Jorge Drews-Jr e professora Dr. Sílvia Silva da Costa Botelho. Agradeço aos integrantes do grupo NAUTEC, mantido pela professora Sílvia Botelho e pelo professor Paulo Drews. Em especial aos colegas Matheus Machado dos Santos, por inúmeras contribuições relativas ao estudo do problema abordado neste trabalho e aos colegas Joel Felipe de Olivera Gaya, Cristiano Steffens, Églen Protas e Felipe Codevilla Moraes por inúmeras discussões acerca do tema de Aprendizagem de Máquina. Agradeço também aos professores Alessandro de Lima Bicho e Adriano Wehrli por todo o suporte durante o período inicial do Mestrado.

"Pedi, e vos será concedido; buscai, e encontrareis;
batei, e a porta será aberta para vós.
Pois todo o que pede recebe; o que busca encontra;
e a quem bate, se lhe abrirá."
(Bíblia Sagrada, Mateus 7, 7-8)

Resumo

Sonares de imageamento frontal (FLSs) são sensores de percepção subaquática que não são afetados pela turbidez. São empregados para auxiliar Veículos Operados Remotamente (ROVs) nas tarefas de exploração, navegação e mapeamento de regiões. Apesar das vantagens do uso de imagens acústicas sobre as imagens ópticas, as primeiras possuem inúmeros desafios inerentes à sua aquisição e representação. Algoritmos de Visão Computacional clássicos possuem restrições quando aplicados a imagens acústicas. O presente trabalho tem como objetivo propor um método para o problema de comparar duas cenas subaquáticas a partir de imagens acústicas obtidas por FLS, avaliando o par de imagens quanto às suas similaridades. São descritas e comparadas algumas das principais abordagens de aprendizagem de profunda para o problema. Dentre elas foi proposta uma arquitetura de regressão para análise da cena, que foi comparada com um método desenvolvido especificamente para comparação de cenas a partir de imagens FLS. Na comparação descrita, a arquitetura de regressão de similaridade proposta obteve melhores resultados. Também foi proposta uma nova estratégia de extração de características para imagens de FLS usando aprendizagem de métrica. Esta estratégia foi comparada com outra abordagem estado-da-arte para obtenção de características, também obtendo resultados superiores na tarefa de recuperação de imagens.

Palavras-chaves: Aprendizagem Profunda; Sonares de Imageamento Frontal; Robótica Subaquática; Redes Convolutivas; Extração de características; Aprendizagem de métrica; Correspondência de Imagens; Recuperação de Imagens.

Abstract

Forward-looking sonars (FLSs) are perception sensors unaffected by underwater turbidity. FLS are used in Remotely Operated Vehicles (ROVs) to help them in the tasks of exploration, navigation and region mapping. Besides the advantages of working with acoustic images rather than optical images, the former presents various challenges inherent to their construction. Classic Computer Vision algorithms have many restrictions when applied to acoustic images. This work has as main goal the proposal of a method for the problem of comparing two underwater scenes perceived with FLS acoustic images, evaluating the image pair with respect to their similarities. It was described and compared some of the main deep learning approaches for the problem. One architecture for similarity regression of the underwater scenes was proposed. This novel architecture was compared with a method specifically designed for underwater scene comparison and achieved better results. Also, a new strategy for automatic feature extraction of FLS images was proposed using deep metric learning. This strategy was compared with a new state-of-the-art approach for feature extraction, also achieving superior results in the task of acoustic image retrieval.

Key-words: Deep Learning. Forward Looking Sonars. Underwater Robotics. Convolutional Neural Networks. Metric learning. Image matching. Image retrieval.

Lista de ilustrações

Lista de tabelas

Lista de Abreviaturas e Siglas

ASFM *Acoustic Structure From Motion* - Estrutura Acústica a partir do Movimento

BN *Batch Normalization* ou Normalização de Lotes

COR *Receiver Operating Characteristic* - Característica de Operação do Receptor

CNN *Convolutional Neural Network* - Rede Neural Convolutiva

DBL *Distance Based Logistic*

DBN *Deep Belief Networks* - Redes de Crença Profundas

DBM *Deep Boltzmann Machines* - Máquinas de Boltzmann Profundas

DGPS *Differential Global Positioning System* - Sistema de Posicionamento Global Diferencial

DVL *Doppler Velocity Log*

DL *Deep learning* - Aprendizagem profunda

FLS *Forward Looking Sonar* - Sonar de Varredura Frontal

FP Falso Positivo

FN Falso Negativo

GD *Gradient Descent* - Gradiente Descendente

GDT Grafo Descritor Topológico

GPU *Graphic Processing Unit* - Unidade de Processamento Gráfico

GPS *Global Positioning System* - Sistema de Posicionamento Global

HDF *High-dimensional Features* - Características de Alta Dimensionalidade

IDA *Incremental Data Association* - Associação de Dados Incremental

IMU *Inertial Measurement Unit*

kNN *k-Nearest Neighbor* - k-Vizinhos Mais Próximos

LMNN *Large Margin Nearest Neighbor* - Vizinho Mais Próximo de Margem Larga (tradução livre)

MCC *Matthews Correlation Coefficient* - Coeficiente de Correlação de Matthews

PCM *Phase Correlation Matrix* - Matriz de Correlação de Fases

PR *Precision and Recall* - Precisão e Revocação

ORB *Orientated FAST and Rotated BRIEF*

RNA Rede Neural Artificial

RNN *Recurrent Neural Networks* - Redes Neurais Recorrentes

ROV *Remotely Operated Vehicle* - Veículo Operado Remotamente

SIFT *Scale-Invariant Feature Transform*

SLAM *Simultaneous Localization And Mapping* - Localização e Mapeamento Simultâneos

SMNet *Sonar Matching Network* - Rede de Correspondência para Sonar

SNR *Signal-to-Noise Ratio* - Proporção entre Sinal e Ruído

SURF *Speed-up Robust Features*

SVM *Support Vector Machine* - Máquina de Vetor Suporte

VC Validação Cruzada - *Cross-validation*

VP Verdadeiro Positivo

VN Verdadeiro Negativo

MNIST *Modified National Institute of Standards and Technology* database

CIFAR10 *Canadian Institute For Advanced Research* - 10 classes database

SVHN *Street View House Numbers - dataset*

STL10 STL-10 *dataset*

Lista de símbolos

I_A	Imagem acústica que representa uma cena subaquática chamada de A. Esta imagem pode ser polar ou cartesiana.
$I(X, Y)$	Imagem acústica que representa uma cena subaquática em coordenadas cartesianas.
$I(\theta, \rho)$	Imagem acústica que representa uma cena subaquática em coordenadas polares.
α_{gt}	Limiar escalar de similaridade entre cenas subaquáticas utilizado para decidir para o *ground truth* se um par de cenas é correspondente ou não correspondente.
α_s	Limiar escalar de similaridade entre cenas subaquáticas utilizado na decisão de um classificador se um par de cenas é correspondente ou não correspondente.
$\alpha_{precisão}$	α_s selecionado que representa o maior valor possível de Precisão e Verdadeiro Positivo (VP) de um modelo, para um determinado α_{gt}.
$\alpha_{revocação}$	α_s selecionado que representa o maior valor possível de Revocação de um modelo, para um determinado α_{gt}.
$\alpha_{especificidade}$	α_s selecionado que representa o maior valor possível de Especificade de um modelo, para um determinado α_{gt}.
η	Taxa de aprendizagem de uma rede neural.
$O_{mín}$	Valor mínimo de uma função de similaridade.
$O_{máx}$	Valor máximo de uma função de similaridade.
$S(I_A, I_B)$	Função de similaridade entre um par I_A e I_B de cenas subaquáticas.
$S_c(I_A, I_B)$	Critério utilizado como função de similaridade entre um par I_A e I_B de cenas subaquáticas.
ϵ_{erro}	Erro entre a saída de uma função de similaridade e o critério adotado para treinamento.
β	Momento utilizado para otimizadores baseados em Gradiente Descendente.

L^1	Norma L^1: $D(X,Y) = \|X - Y\|_1$
L^2	Norma L^2: $D(X,Y) = \|X - Y\|_2$

Sumário

1 Introdução

A maior parte da superfície do planeta Terra é coberta por água. O interesse no ambiente subaquático compreende a extração de recursos, exploração de sítios arqueológicos submersos e pesquisas biológicas. A percepção deste ambiente é desafiadora; ao considerar-se aspectos inerentes ao comportamento da luz em meios participativos, a visibilidade do observador é restrita em função dos fenômenos de espalhamento e absorção dos fótons pelas partículas presentes no meio. Uma alternativa é o uso de sensores do tipo Sonar, que trabalham no espectro acústico e operam independente das condições de luminosidade. Sonares são usados em barcos, submarinos e robôs subaquáticos para auxiliar a navegação.

A navegação subaquática com *Remotely Operated Vehicle* - Veículo Operado Remotamentes (ROVs) permite a exploração de cenários inseguros para mergulhadores em função da profundidade. Robôs subaquáticos podem ser empregados em inspeção de casco de navio (HOVER et al., 2012), busca de artefatos e mapeamento de regiões através de mosaico (HURTÓS et al., 2015) e etc. Para utilização de ROVs de forma autônoma, um aspecto fundamental para a navegação é conhecer a sua localização.

Os sensores utilizados para localização em ambientes terrestres são diferentes dos sensores utilizados em ambientes subaquáticos. O uso de *Global Positioning System* - Sistema de Posicionamento Global (GPS) em ambiente aquático, por exemplo, só é possível em sua superfície. Este fenômeno ocorre devido às ondas de rádio serem rapidamente absorvidas pela água (MACHADO; DREWS-JR; BOTELHO, 2016). Uma alternativa é o uso de sensores de odometria como a *Inertial Measurement Unit* (IMU) e o *Doppler Velocity Log* (DVL) para mensurar a trajetória (AULINAS et al., 2011). Considerando que sensores de odometria apresentam alguma forma de ruído, normalmente são aplicados filtros para tratamento do sinal (e.g. filtro de Kalman (SORENSON, 1985)). Algoritmos de *Simultaneous Localization And Mapping* - Localização e Mapeamento Simultâneos (SLAM) subaquáticos também são empregados para estimar a posição e mapear o caminho percorrido pelo robô (RIBAS et al., 2006) (GUTH et al., 2013).

Um subproblema de SLAM é a detecção de fechamento de *loops*. A detecção consiste em decidir se um robô já visitou ou não uma determinada localidade em sua trajetória. Algoritmos para resolução de detecção de *loop* lidam com entradas de sensores inerciais e visuais. Estes algoritmos fazem uso tanto de estimativas de posição e orientação, quanto extração de características para análise da cena (HO; NEWMAN, 2006). Algumas referências são escolhidas para corrigir a estimativa de trajetória. Em especial, com a popularização dos *Forward Looking Sonars* (*Forward Looking Sonar* - Sonar de

Varredura Frontal (FLS)) ou sonares de imageamento frontal é possível obter imagens acústicas com grande alcance e resolução que podem ser utilizadas como referências.

Além de aplicações em SLAM, um dos problemas fundamentais da Visão Computacional é a Correspondência de cenas, que é responsável por comparar e estabelecer correspondências entre cenas, lugares, ou partes de imagens. Abordagens clássicas de Visão Computacional dependiam de detectores de *features* como *Speed-up Robust Features* (SURF)(BAY; TUYTELAARS; GOOL, 2006) ou *Scale-Invariant Feature Transform* (SIFT)(LOWE, 1999) para representar imagens ou cenas. O uso de features ensejou o avanço de algoritmos de classificação, detecção de objetos, reconhecimento de faces entre outros. Entretanto, sensores do tipo FLS possuem particularidades que os diferenciam das câmeras, exigindo cuidados especializados no projeto de algoritmos.

1.1 Sonar de Imageamento Frontal

Sonares de imageamento frontal ou *Forward Looking Sonars* (FLS) são dispositivos emissores de ondas acústicas. As ondas são propagadas até colidirem com um obstáculo ou serem absorvidas completamente. Parte da energia destas ondas é refletida, sendo armazenada por um conjunto de hidrofones. As ondas capturadas pelos hidrofones são organizadas de acordo com a sua direção de retorno e a sua distância em relação ao objeto refletor. Esta informação é estimada de acordo com a diferença do tempo de captura de cada hidrofone para a mesma onda, levando em consideração a velocidade do som na água. Uma imagem acústica cartesiana, $I(X, Y)$, em formato de leque, é uma das formas de se representar as informações retidas pelo sonar durante um período de tempo Neste tipo de imagem os valores de intensidade dos pixels são associados com o retorno armazenado sobre o tempo. Os pixels ficam indexados conforme sua distância r e direção azimutal θ do sonar, observado na Figura 1 a). Há também a representação polar, observável nas Figuras 2(e) e 2(f).

FLSs podem variar nas suas características de abertura do campo de visão, alcance e freqüência de captura das imagens acústicas dependendo da sua aplicação (MACHADO; DREWS-JR; BOTELHO, 2016). De forma geral, um FLS pode ser usado para inspeção de casco de navio (HUANG; KAESS, 2016), desvio de objetos(MACHADO; DREWS-JR; BOTELHO, 2016), auxílio na navegação(SILVEIRA et al., 2015) e mosaico (HURTÓS et al., 2015). O DIDSON por exemplo possui um alcance menor que o Blueview P900-130, mas sua maior freqüencia de operação pode ser interessante para pesquisadores interessados em inspeção de objetos próximos.

Nas figuras: Fig. 1 b) e Fig. 1 c) são demonstradas algumas das distorções decorrentes da forma de representação em leque. Apesar das vantagens provenientes dos FLS, a imagens acústicas apresentam algumas características inerentes à sua construção e à

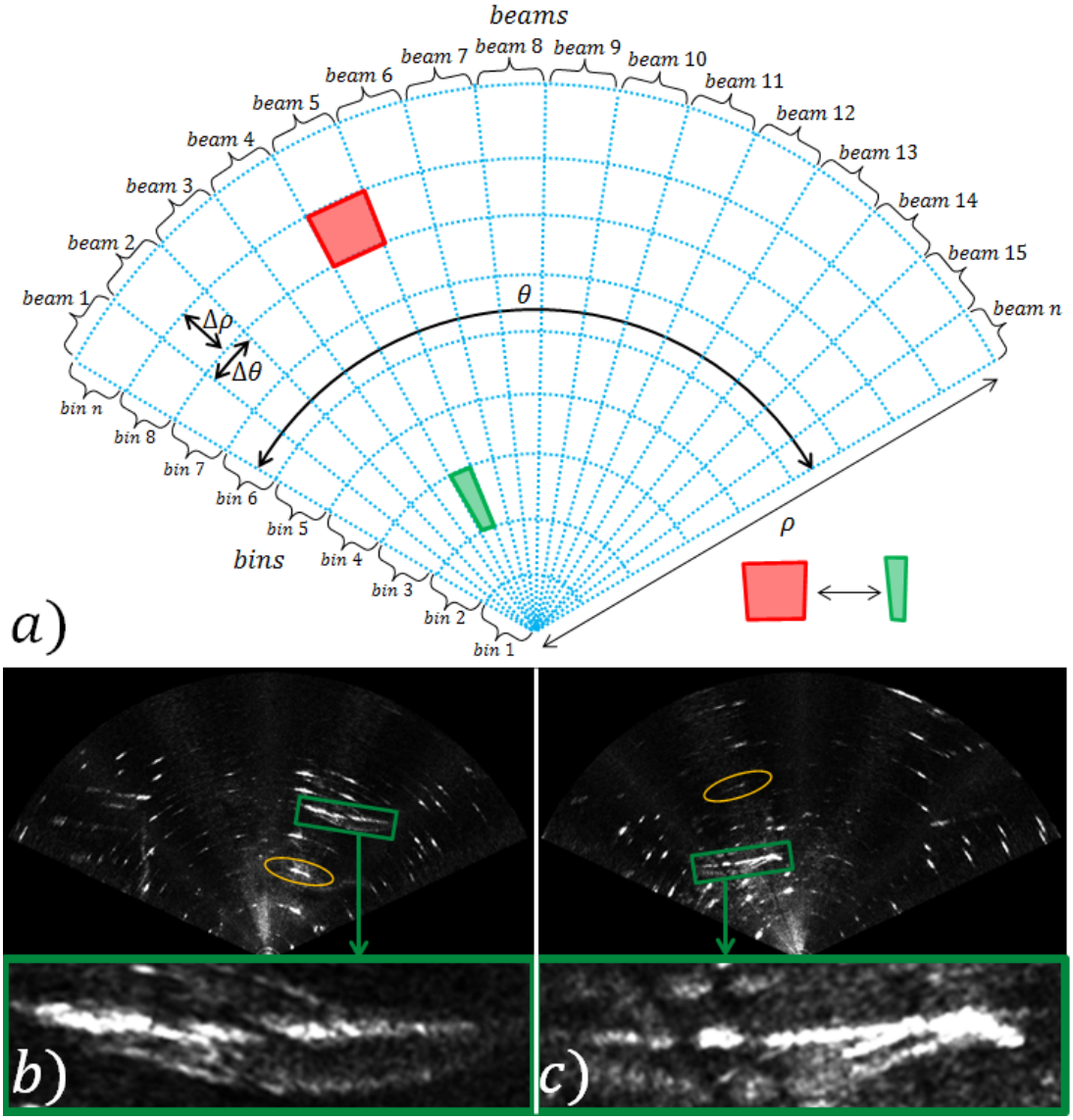


Figura 1: Imagens Acústicas: a) exemplo de uma aquisição de um FLS, b) e c) imagens adquiridas de diferentes posições. Caixas verdes representam as distorções de forma de um mesmo objeto. A elipse amarela na figura b) quase desaparece em c) devido ao efeito da sombra acústica criado pela forma destacada em verde. Imagem retirada de (MACHADO; DREWS-JR; BOTELHO, 2016).

propriedades das ondas acústicas:

- **Resolução não-homogênea** - Os *bins* próximos cobrem uma área menor e possuem menos pixels que *bins* distantes.

- **Distorção Acústica** - A distorção acústica causa um achatamento dos objetos, dificultando o reconhecimento de seu formato.

- **Projeção em 2D** - A elevação dos retornos acústicos não é dinstingüível em função da construção dos FLSs. Isto gera um problema de ambigüidade que afeta a estimativa de distância dos *bins*(JOHANNSSON et al., 2010).

- **Intensidades de Pixel não-uniformes** - Este comportamento se relaciona com a atenuação da água, o *tilt* do sonar e diferença de sensitivdade entre os hidrofones. A sensitividade dos hidrofones também pode ser afetada por sujeira.

- **Reverberação Acústica** - Ocorre quando uma onda acústica é refletida múltiplas vezes por um objeto, gerando múltiplas réplicas de um mesmo objeto na imagem.

- **Sombra acústica** - É caracterizada por uma região preta depois de um objeto que esconde parte da cena capturada. O reposicionamento do sonar causa movimento das sobras acústicas e muda a região da imagem que sofre oclusão.

- **Ruído** - Em função da baixa proporção de *Signal-to-Noise Ratio* - Proporção entre Sinal e Ruído (SNR), imagens acústicas são muito ruidosas. Os ruídos podem ocorrer pela reflexão da superfície da água; por mútua interferência causada por speckle; pelos motores do robô ou ainda por veículos próximos.

É possível encontrar na literatura de Visão Computacional muitos trabalhos sobre *matching* ou correspondência de imagens ópticas. Entretanto, é conhecido o fato de que imagens ópticas subaquáticas apresentam visibilidade reduzida devido a turbidez. Apesar de existirem trabalhos que utilizem abordagens clássicas de Visão Computacional baseadas em extração de características com imagens acústicas subaquáticas para navegação, correspondência e mosaico, Hurtós et al. (2013) afirma que estes trabalhos baseados em extratores de características clássicos lidam com pares de imagens com diversas restrições de ambiente, pequena diferença de ponto de vista e curto intervalo de tempo entre as imagens, tendo assim uma baixa repetibilidade. Neste contexto, a criação de um método capaz de lidar com essas variações de ponto de vista entre cenas e lidar com quadros muito distantes temporalmente é um problema a ser levado em consideração.

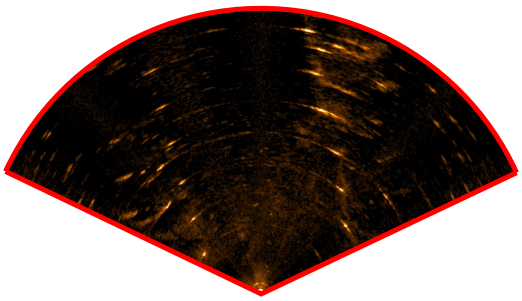
1.2 Descrição do Problema

O problema a ser tratado neste texto é calcular a similaridade de duas cenas subaquáticas representadas por duas imagens acústicas I_A, e I_B de um único canal, obtidas pelo mesmo modelo de FLS com as mesmas configurações de parâmetros de ângulo de abertura e profundidade através de uma função $S(I_A, I_B)$. Portanto, é necessário encontrar funções $S(I_A, I_B)$ que calculem o grau de similaridade dessas imagens na literatura. Estas funções devem considerar que as imagens capturadas podem conter poucos elementos estruturados. Também é possível que os objetos capturados pelos *beams* do FLS sejam dinâmicos, como peixes; ou semi-estáticos, como barcos que podem estar em um lugar, mas não necessariamente estarão lá dado um grande intervalo de tempo. De forma análoga, este problema está relacionado ao Reconhecimento de Lugares representados por imagens

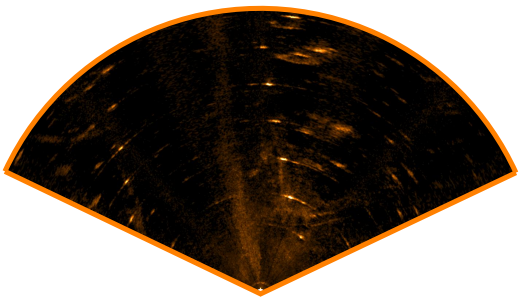
ópticas. É importante notar que para imagens ópticas, o problema de reconhecimento de lugares ainda é um problema em aberto (LOWRY et al., 2016).

(a) Correspondência Entre Cenas

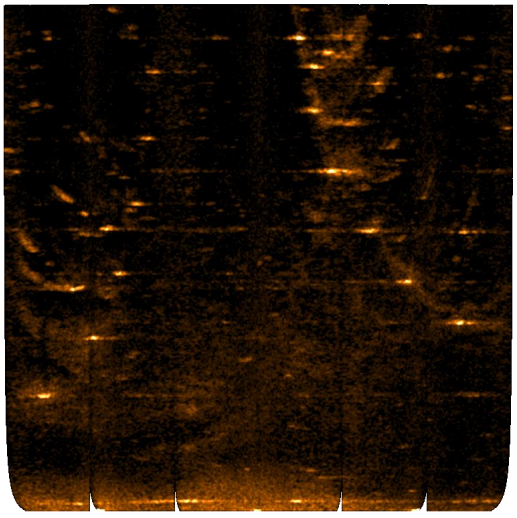
(b) Yacht Club



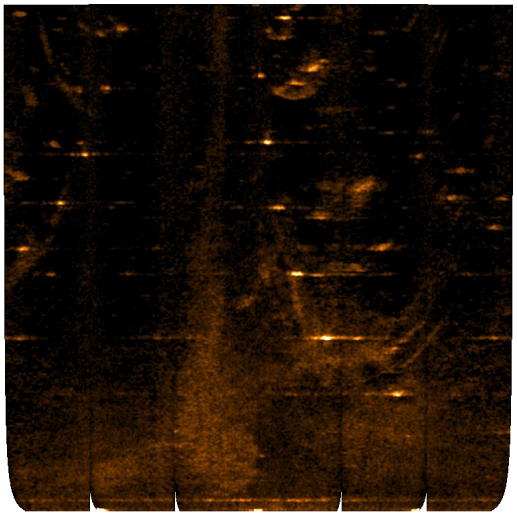
(c) I_A Cartesiana



(d) I_B Cartesiana



(e) I_A Polar



(f) I_B Polar

Figura 2: Imagens Acústicas: A Figura 2(a) mostra duas imagens acústicas que capturam elementos de uma mesma cena. A Figura 2(b) mostra o local de captura das imagens acústicas. As Figuras 2(c) e 2(d) mostram imagens acústicas em formato *I(X, Y)* enquanto as Figuras 2(e) e 2(f) imagens acústicas em formato *I(θ, ρ)*. Dadas duas cenas subaquáticas capturadas por Sonar é necessário avaliar o grau de similaridade entre as duas.

1.3 Aprendizagem Profunda

Deep learning - Aprendizagem profunda (DL) é um nicho de algoritmos e técnicas que é resultado de um avanço extenso e contínuo de inúmeros pesquisadores de aprendizagem de máquina. Schmidhuber (2015) destaca similaridades e diferenças entre arquiteturas de redes neurais do milênio passado (e.g. *Neocognitron*), com as redes neurais atuais. Atualmente, os modelos de DL ganharam destaque em competições de Visão Computacional dentre alguns fatores, pela sua acurácia estado-da-arte. Um dos modelos responsáveis por atrair a atenção da comunidade científica para este tipo de rede novamente foi a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), sendo o primeiro modelo de *Convolutional Neural Network* - Rede Neural Convolutiva (CNN) a obter resultados estado-da-arte no ILSVRC-2012. O modelo utilizava todo o potencial das GPUs de processamento genérico, sendo necessária a divisão do modelo em duas *pipelines* de processamento para duas placas de vídeo em função do seu tamanho.

Redes de aprendizagem profundas podem servir para múltiplos tipos de problemas. Problemas como classificação de vídeos(KARPATHY; FEI-FEI, 2015), segmentação (BADRINARAYANAN; KENDALL; CIPOLLA, 2017), tradução imagem para imagem (ZHU et al., 2017), estimativa de *pose*(KENDALL; GRIMES; CIPOLLA, 2015) e redução dimensional (LI et al., 2016). Em virtude do amplo espectro de aplicação dessas técnicas, é natural questionar como elas poderiam ser aplicadas para extrair características relevantes de imagens de FLS para comparar duas cenas subaquáticas.

1.3.1 Redes Neurais

Dentro do nicho de algoritmos de Aprendizagem de Máquina, técnicas podem ser subdivididas quanto à necessidade de supervisão em supervisionadas e não supervisionadas. Na aprendizagem supervisionada, um modelo precisa ser treinado previamente com um conjunto de dados que contenha um par (E, S) onde E representa entrada e S a saída desejada. A idéia é que o modelo seja calibrado com um **conjunto de dados de treinamento**, tornando-se capaz de realizar inferências precisas sobre entradas de dados do mundo real.

Redes neurais são modelos matemáticos de aprendizagem de máquina que originalmente buscavam mimetizar o processo de aprendizagem humana através de neurônios artificiais e suas interconexões. Cada conexão de um neurônio de uma Rede Neural Artificial (RNA) possui conexões com um conjunto de entradas, uma ou mais camadas internas e uma camada de neurônios de saída (HAYKIN, 2007). Para simular o estímulo das conexões, são utilizadas funções matemáticas não-lineares chamadas de **funções de ativação** (e.g. $tanh(x)$, sigmoidal). As conexões possuem **pesos sinápticos** (ou simplesmente **pesos**), que são valores escalares utilizados para calibrar a rede. A entrada de um

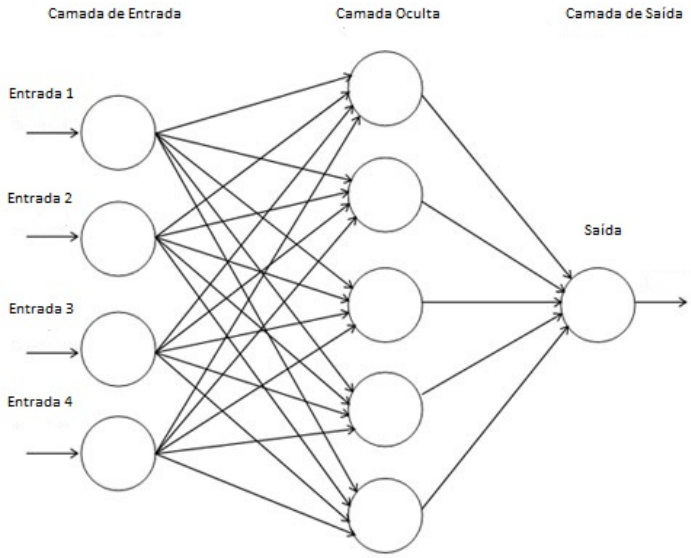


Figura 3: Rede Neural: Exemplo de uma rede neural com três camadas (Adaptada de (CAZALA, 2017)). Este exemplo de rede neural possui quatro entradas e cinco neurônios intermediários. O resultado da rede é apenas um valor numérico por possuir apenas um neurônio de saída.

neurônio é chamado de **campo local induzido**. O campo local induzido é uma operação matemática que varia conforme a camada de neurônios em que o neurônio está situado. Na arquitetura Perceptron de Múltiplas Camadas, ilustrada na Figura 3, cada neurônio da primeira camada possui como campo local induzido o somatório ponderado de suas entradas. Nas camadas intermediárias, o campo corresponde ao somatório ponderado das conexões neurônios da camada anterior.

1.3.2 Redes Neurais Convolutivas

Convolutional Neural Network ou Redes Neurais Convolutivas CNNs diferem das Redes Neurais convencionais quanto ao tipo de operação realizada para alimentar uma função de ativação. Ao invés de utilizar uma função de soma ponderada da entrada com os pesos, é realizada uma convolução discreta da entrada com um *kernel* (núcleo) de pesos. LeCun et al. (1998) propôs a arquitetura CNN em sua rede LeNet para o reconhecimento de caracteres. Do ponto de vista de Visão Computacional, as camadas convolutivas fornecem uma alternativa para obtenção de *features* ou características da imagem, já que os mapas de *features* são aprendidos durante o treinamento da Rede. Estes mapas são empregados como uma alternativa em relação ao uso de métodos clássicos de extração de características como SIFT (LOWE, 1999). Zagoruyko, por exemplo, compara o uso de CNN com arquiteturas usando SIFT para realizar *matching* de *patches* de imagens (ZAGORUYKO; KOMODAKIS, 2015). Na Figura 4 é possível observar um diagrama de uma arquitetura convolutiva.

A nomenclatura *Deep Neural Network* ou Rede Neural Profunda refere-se a um

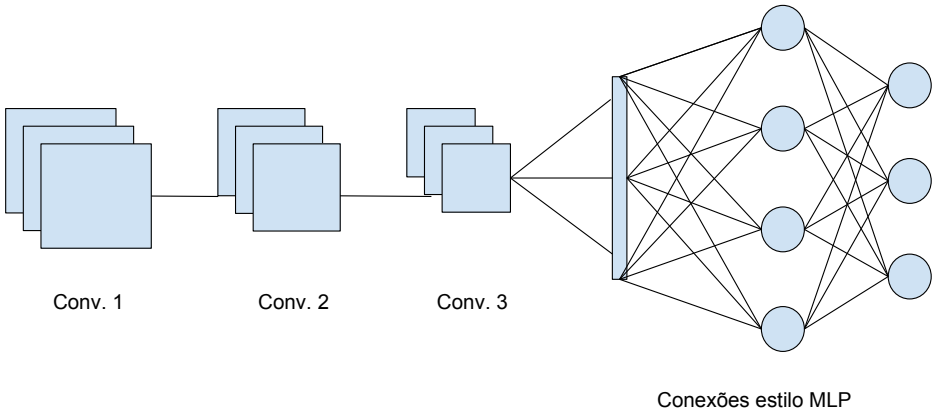


Figura 4: Rede Neural Convolutiva: Exemplo de uma rede neural convolutiva com três camadas de convolução. Técnicas de subamostragem são aplicados sobre os mapas de características. Após a entrada ser processada pelas convoluções, a saída da camada Conv. 3 alimenta uma Rede Neural com conexões completamente conectadas (HAYKIN, 2007).

maior número de camadas de pesos. Algumas arquiteturas podem chegar a 152 camadas de profundidade (HE et al., 2015). Entretanto, arquiteturas com 19 camadas já podem ser consideradas muito profundas (SIMONYAN; ZISSERMAN, 2014). Redes profundas não se restrigem a abordagens convolutivas. *Deep Belief Networks* - Redes de Crença Profundas (DBN), *Deep Boltzmann Machines* - Máquinas de Boltzmann Profundas (DBM), *Recurrent Neural Networks* - Redes Neurais Recorrentes (RNN) também podem ser consideradas técnicas de aprendizagem profunda.

1.4 Objetivo Geral

O objetivo deste trabalho é o estudo e aplicação de técnicas de aprendizagem profunda para cálculo de similaridade de cenas subaquáticas representadas por duas imagens acústicas de escala única I_A e I_B obtidas por um FLS. Uma premissa é que I_A e I_B sejam capturadas com os mesmos parâmetros de configuração de um mesmo modelo de FLS.

1.5 Objetivos Específicos

O sucesso do objetivo geral depende tanto da aplicação em que o método proposto será avaliado, quanto a estratégia adotada pelo método. Os objetivos específicos deste trabalho são:

- Revisar técnicas de aprendizagem profunda na literatura que possam ser aplicadas neste trabalho.

- Propor um método de aprendizagem profunda para comparar duas cenas subaquáticas representadas por imagens de FLS.
- Avaliar o desempenho da arquitetura proposta com imagens acústicas reais coletadas por um ROV.
- Revisar possíveis métricas de avaliação para os métodos propostos.

1.6 Organização do Trabalho

O texto deste trabalho está organizado em:

- O Capítulo 2 revisa alguns dos trabalhos relacionados ao problema.
- O Capítulo 3 discute possíveis modelagens para o problema e propõe alternativas para atender ao objetivo geral.
- O Capítulo 4 apresenta experimentos realizados durante o desenvolvimento desta dissertação.
- E por fim o Capítulo 5 resume pontos destacados no trabalho, enquanto que o Capítulo estabelece algumas direções para trabalhos futuros.

1.7 Contribuições

Esta dissertação apresenta duas abordagens para comparação de cenas subaquáticas correspondentes a partir de imagens de acústicas. A primeira abordagem é uma **Arquitetura de Regressão de Escore de Similaridade**. A arquitetura é descrita e comparada com um método desenvolvido especificamente para sonares de imageamento frontal. A segunda abordagem utiliza de uma arquitetura já conhecida com um **Pipeline de Treinamento com *Triplets***. A segunda abordagem foi comparada com uma implementação do *pipepline* de treinamento de um método desenvolvido para inspeção de cascos de navio (LI et al., 2016). Uma vantagem do *pipeline* proposto neste trabalho foi que a nova abordagem permite desacoplar a similaridade entre imagens da anotação geográfica direta da cena, permitindo que o método obtenha resultados superiores em ambiente desconhecido.

Publicações realizadas durante a pesquisa:

- RIBEIRO, P. O. C. de S. et al. Underwater place recognition in unknown environments with triplet based acoustic image retrieval. 17th IEEE International Conference on Machine Learning and Applications (Submetido). 2018.

- PROTAS, E. et al. Visualization techniques applied to image-to-image translation. Brazilian Conference on Intelligent Systems - BRACIS. 2018.
- RIBEIRO, P. O. C. d. S. et al. Forward looking sonar scene matching using deep learning. In: IEEE. *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. [S.l.], 2017. p. 574–579.
- MACHADO, M. et al. Description and matching of acoustic images using a forward looking sonar: A topological approach. *IFAC-PapersOnLine*, Elsevier, v. 50, n. 1, p. 2317–2322, 2017.
- SANTOS, M. dos et al. Object classification in semi structured enviroment using forward-looking sonar. *Sensors*, v. 17, n. 10, 2017. ISSN 1424-8220. Disponível em: <http://www.mdpi.com/1424-8220/17/10/2235>.

2 Trabalhos Relacionados

Existem diversos trabalhos que utilizam FLS como principal sensor em diferentes aplicações de robótica. Existem ainda mais trabalhos que utilizam de aprendizagem de máquina para resolver problemas de Visão Computacional utilizando câmeras ópticas. Apenas um número menor pesquisadores trabalham com aprendizagem de máquina voltada para o processamento de imagens acústicas de FLS. Portanto, primeiro será feita uma breve introdução de problemas de processamento de imagens acústicas oriundas de FLS. Em segundo lugar, serão apresentadas técnicas de aprendizagem de máquina relevantes a esta dissertação. Em seguida, diferentes arquiteturas cujos propósitos se assemelham ao problema aqui tratado serão apresentadas; ainda que o principal sensor de percepção seja diferente. E por fim, serão apresentados trabalhos que utilizam de técnicas de aprendizagem profunda com FLS.

2.1 FLS e Percepção Subaquática

Considerando-se SLAM baseado em sonar, Johannsson et al. (2010) apresenta uma solução de minimização de *drift*. A solução consiste na extração de *features* densas da imagem acústica e aplicação do algoritmo *Normal Distribution Transform* (NDT) como função $S(I_A, I_B)$ para encontrar o alinhamento entre um par de imagens. Esta informação é combinada com dados de velocidade de um DVL, a atitude e a aceleração mensurados pela IMU. A fusão destes dados é utilizada para obter uma estimativa melhorada da trajetória do robô. O trabalho é avaliado através do erro entre o *ground truth* (que representa uma trajetória correta), o *dead reckoning* (estimativa de trajetória inferida diretamente pelos sensores) e a trajetórias estimada pelo método. Não há uma avaliação específica sobre correspondência de imagens acústicas. Além disto, no presente trabalho há o interesse em soluções que utilizem apenas dados provenientes do FLS por este ser sensor de percepção comumente acoplado a ROVs pequenos.

Em Hurtós et al. (2015) é proposto um *pipeline* de registro de imagens FLS para construção de um mosaico 2D. Motivados por problemas de sonar como ruído, baixa resolução e iluminação não-uniforme, as imagens são convertidas para o domínio da freqüência. A primeira etapa é a obtenção da estimativa de rotação. A estimativa é calculada utilizando a *Phase Correlation Matrix* - Matriz de Correlação de Fases (PCM) com as duas imagens no formato polar. O ΔX da correlação representa o *yaw* ou rotação θ no plano. Obtida a estimativa de rotação, uma imagem cartesiana do par é rotacionada, e a estimativa de translação é calculada usando novamente a PCM. Com a matriz de transformação calculada, é possível alinhar as duas imagens. Pode-se afirmar que a PCM é utilizada

como $S(I_A, I_B)$ entre os pares de imagens. Resultados superiores para o alinhamento de imagens são encontrados em relação à resultados comparados com abordagens baseados no domínio espacial. O alinhamento possui resultados melhores quando otimizado *offline*. Outro aspecto a ser considerado é o fato do método de registro lidar com imagens relativamente próximas, ou praticamente consecutivas. Não são feitas considerações a respeito de imagens muito distantes em temporalidade ou com grandes distorções em função da orientação do ROV durante a captura.

Huang e Kaess (2016) continua o trabalho anterior de *Acoustic Structure From Motion* - Estrutura Acústica a partir do Movimento (ASFM) inserindo o *Incremental Data Association* - Associação de Dados Incremental (IDA). O algoritmo dos autores utiliza uma árvore de correspondências que depende de uma extração de *features*. Uma nova *landmark* é adicionada caso não exista nenhuma correspondência na árvore. A acurácia do IDA-ASFM mensurada pelo seu reconhecimento de *features* em simulação sem a inclusão de *features* espúrias para criação de ruído é 90.9%, significando que o algoritmo encontra a associação de dados correta 90.9% do tempo. Esta acurácia decai para 72.4% com maiores simulações de ruído e medições espúrias. É conduzido um experimento com cinco *frames* diferentes de um objeto estruturado em um tanque (uma escada), e o uso de um sonar DIDSON, onde eles efetivamente conseguem detectar os marcadores para a associação. O trabalho lida com a geometria em três dimensões. O trabalho não resolve completamente o problema de associar no mapa de lugares as novas estruturas descritas pelas características. É razoável esperar que o desempenho do método caia em cenários não-estruturados e desconhecidos como uma caverna subaquática ou um terreno irregular.

Machado et al. (2016) propõe um novo descritor topológico para o problema de detecção de *loops* de SLAM usando apenas imagens acústicas de um FLS. Este método será chamado de GDT. O método realiza uma extração de *features* densa usando uma nova abordagem de segmentação baseada em intensidades de pico. Os segmentos extraídos são descritos por um grafo de funções Gaussianas que representam o formato do objeto e seu relacionamento topológico. A detecção de *loops* é feita entre pares de imagens ao comparar os grafos topológicos. A $S(I_A, I_B)$ é realizada através de uma busca de similaridades em grafos através de vértices correspondentes. Utiliza-se uma heurística para tratamento de ambigüidades de dois ou mais vértices candidatos a correspondentes. Uma avaliação é feita em termos de correspondência de nodos, em que grafos resultantes de imagens acústicas sobrepostas são comparados. O método encontra correspondência em todos os 35 testes de detecção de *loop*.

É possível encontrar na literatura trabalhos que utilizam o FLS como principal sensor de percepção. Em virtude das propostas se utilizarem de técnicas e abordagens distintas para comparação de imagens acústicas, a definição de uma função de similaridade $S(I_A, I_B)$ é uma forma de agrupar trabalhos que são especializados em subproblemas

diferentes de um problema maior. Os subproblemas podem ser: encontrar um deslocamento físico entre dois *frames*; encontrar a correspondência de características semânticas descritoras das imagens; busca de correspondências visuais entre as imagens; ou associar as imagens em um grafo que represente o mundo físico a ser observado. Enquanto que traçando um paralelo entre todos os trabalhos, pode-se dizer que eles encontram uma função $S(I_A, I_B)$ capaz de encontrar uma relação entre as imagens.

De forma geral, os trabalhos mencionados não resolvem completamente problema da similaridade entre as cenas. Com a premissa de que os *frames* coletados sejam próximos, é possível realizar uma estimativa de rotação e translação. Entretanto, isto ainda representa uma limitação da abordagem ao lidar com grandes diferenças de perspectiva entre as duas imagens acústicas. O IDA-ASFM tenta relacionar objetos sem a restrição bidimensional. Contudo, mesmo em simulação o método não resolve completamente o problema. Quanto ao seu teste, o método foi validado apenas com um conjunto muito pequeno de imagens reais. Isto reafirma a necessidade de um método robusto capaz de realizar uma comparação das cenas ainda que as diferenças entre as imagens sejam difíceis de serem percebidas.

2.2 Arquiteturas de Redes Convolutivas

Nesta seção foram selecionadas algumas das principais arquiteturas CNN que se relacionam a problemas análogos ao definido na Introdução. O escopo de trabalhos relacionados foi ampliado nesta direção com o intuito de estudo de técnicas que possam aplicadas em diversas representações de cenas. Uma mesma rede de classificação (SZEGEDY et al., 2017), por exemplo, pode ser aplicada para diferentes *datasets* sendo necessário apenas retreinar aquela arquitetura através de transferência de conhecimento ou *transfer learning*. Em casos de diferença do número de classes, pode-se ajustar apenas o final da arquitetura trocando a dimensão de saída da rede. Sendo assim, é importante conhecer tanto da capacidade de aprendizagem de diferentes arquiteturas, técnicas de treinamento e escolha de dados para conjuntos de treinamento para garantir que a rede é capaz de desempenhar uma determinada tarefa.

2.2.1 Reconhecimento de Locais

Kendall, Grimes e Cipolla (2015) usam uma adaptação da GoogLeNet(SZEGEDY et al., 2015) em um sistema de estimativa de 6 graus de liberdade em Cambridge. Dada uma imagem do local, o sistema deles é capaz de estimar uma regressão da *pose* original $\hat{p}$ usando imagens da mesma cena em diferentes condições climáticas, ou capturadas com câmeras de diferentes distâncias focais. Sunderhauf et al. (2015) propõe um sistema de Reconhecimento de Locais para cenários ao ar livre. O sistema utiliza da arquiteura

AlexNet(KRIZHEVSKY; SUTSKEVER; HINTON, 2012) para extração de *features* das paisagens; de projeção aleatória Gaussiana para redução de dimensionalidade e busca de vizinho mais próximo usando *cosine distance* entre os descritores para cálculo de escore. Apesar da natureza das imagens serem diferentes, algoritmos para reconhecimento de locais também lidam com efeitos e elementos que variam com o tempo (LOWRY et al., 2016).

2.2.2 *Image Matching* ou Correspondência de Imagens

Zbontar e LeCun (2016) foram capazes de estimar a disparidade entre imagens adquiridas com par stereo e obter a correspondência entre elas com o uso de uma rede de arquitetura siamesa. As etiquetas das imagens utilizadas no *dataset* deles foram construídas a partir do *ground truth* obtido de uma câmera stereo. Zagoruyko e Komodakis (2015) discutem variações da arquitetura siamesa para CNNs que realizam *matching* de *patches* de imagens. Eles também apresentam os resultados de uma arquitetura de dois canais de entrada. Este tipo de arquitetura assemelha-se a uma rede de classificação, onde cada canal de entrada é alimentado com uma imagem representa em um único canal de cor. Esta abordagem é capaz de computar relações entre o par de imagens desde a primeira camada. Em seu estudo, eles obtiveram resultados melhores do que os obtidos com os das arquiteturas siamesas.

2.2.3 *Image Retrieval* ou Consulta de Imagens baseada em Conteúdo

Uma alternativa para o uso de redes siamesas é introduzido por Hoffer e Ailon (2015). O tipo de rede proposto pelos autores aprende representações de imagens através do uso de comparações de distância das representações. A rede *Triplet* é projetada com a finalidade de ser aplicada em ranking e recuperação de imagens. Vo e Hays (2016) descrevem e comparam CNNs siamesas, redes de classificação e redes *Triplet* para a tarefa de geolocalização. Os autores propuseram novas funções de erro para as redes siamesas e *Triplet*. Durante o estudo, eles também observaram que estimar a rotação em conjunto com a posição aumentou a acurácia da regressão de posição. Este resultado reforça observações feitas por (KENDALL; CIPOLLA, 2017) em uma versão mais recente da *PoseNet*.

2.2.4 Redes Convolutivas que utilizam imagens de FLS

CNNs são capazes de extrair mapas de *features* úteis para encontrar relações geométricas entre imagens ópticas. Estas *features* são robustas à efeitos externos, como diferenças no ambiente e condições de iluminação. Apesar das diferenças entre imagens acústicas e ópticas, CNNs apresentam características que poderiam ser aplicadas para atender aos desafios das imagens acústicas.

Valdenegro-Toro (2017) explora o uso de arquiteturas baseadas em *matching* de *patches* no trabalho de Zagoruyko e Komodakis (2015). Os experimentos consideram a classificação dos pares de *patches* de diferentes objetos submersos em um tanque de água. O sonar utilizado é o ARIS Explorer 3000. Os resultados comparam redes siamesas e um classificador de dois canais com abordagens clássicas. As suas implementações baseadas em CNN obtém resultados melhores que extratores de características clássicos (e.g. SURF, SIFT, *Orientated FAST and Rotated BRIEF* (ORB)(RUBLEE et al., 2011) e etc) e outras abordagens para classificadores (e.g. *Support Vector Machine* - Máquina de Vetor Suporte (SVM) e *Random Forest*).

Utilizando FLS como sensor de percepção, Li et al. (2016) propôs o uso de *feature maps* aprendidos por uma CNN de regressão dimensional para imagens acústicas subaquáticas. A rede dos autores é uma adaptação da arquitetura GoogLeNet, usando um vetor de pesos da penúltima camada como descritores. A rede dos autores é treinada para predizer a posição (x, y, z) das imagens de sonar capturadas de uma inspeção de casco de navio. A posição dos frames usada como *label* de treinamento foi obtida através de um *bundle adjustment*. A penúltima camada de alta dimensão é nomeada *local-preimage* e é usado como observação em um filtro de partículas junto com sensores de odometria para estimativa de localização.

2.3 Conclusão

Foi observado com os trabalhos anteriores que: I) CNNs conseguem recuperar relações geométricas de imagens ópticas com diferenças drásticas. II) Abordagens baseadas em CNN são capazes de corretamente associar imagens de um FLS para *labels* anotadas. III) Até onde o autor desta dissertação tem conhecimento, a abordagem de *Triplets* não foi explorada para atender o problema de encontrar similaridades em imagens de FLS. Portanto, na seção seguinte são descritas algumas possíveis modelagens para o problema utilizando de técnicas recentes em aprendizado profundo.

3 Metodologia

Para atender ao problema da comparação entre cenas, pretende-se utilizar técnicas de aprendizagem de máquina profunda. Em especial as CNNs, que são capazes de aprender os mapas de características mais relevantes para uma determinada tarefa. Estes mapas de características funcionam de forma análoga à descritores clássicos como SURF(BAY; TUYTELAARS; GOOL, 2006) ou SIFT(LOWE, 1999). A maior parte das CNNs estado-da-arte em problemas de Visão Computacional que foram estudadas para confecção deste trabalho foram supervisionadas, por este motivo, decidiu-se utilizar de dados anotados previamente para treinamento das CNNs.

Durante o andamento desta dissertação, definições como qual o tipo de saída da rede, qual a anotação de dados para treinamento e qual tipo de arquitetura seriam utilizadas foram alterados de forma iterativa. A Seção 3.1 discute algumas das possíveis modelagens para o problema que foram de fato implementadas. A Seção 3.2 define a implementação de uma nova arquitetura de regressão para imagens de FLS. A Seção 3.3 define a implementação de uma nova arquitetura profunda de aprendizagem de métrica para comparação imagens de FLS.

3.1 Modelagem do Problema

3.1.1 Medida de Similaridade

Para atender ao problema de mensurar a similaridade entre cenas capturadas por FLS é necessário uma forma de anotar elementos em comum entre as imagens. Analisar e anotar manualmente cada imagem acústica por um especialista poderia oferecer mais descrições para comparar posteriormente cada par. Entretanto, não seria escalável para uma grande quantidade de dados anotar cada possível par de imagens. Os *datasets* utilizados na Seção 3.4, contém dezenas de milhares de imagens. Se existisse uma forma automática de anotar grandes quantidades de dados, isto permitiria um melhor treinamento para uma arquitetura de aprendizagem de máquina. Com este intuito, os dois *datasets* utilizados de dados reais neste trabalho utilizaram um GPS e a bússola do ROV para anotar os dados. Estas leituras de sensores são úteis para fins de anotação, mas não é possível orientar um ROV utilizando GPS em virtude da tecnologia não ser aplicável a ambientes subaquáticos (MACHADO; DREWS-JR; BOTELHO, 2016). Assumiu-se neste trabalho que com a posição e orientação dos instantes em que as imagens do FLS foram capturadas é possível recuperar a área bidimensional geométrica das capturas. A área em comum destas capturas, descrita na Figura 5, é uma medida a *priori* do quanto estas imagens

potencialmente capturam elementos em comum. Para diferenças pequenas de orientação e posição relativa entre as imagens esta medida se aproxima da sua totalidade. Considerando a $A_{A \cap B}$ do par há uma medida de avaliação que independe dos fenômenos descritos na Seção 1.1. Rotações altas entre as imagens apresentam inúmeros desafios: causam distorções de perspectiva; podem sofrer diferentes sombras acústicas; sendo um desafio para o processamento deste tipo de imagem.

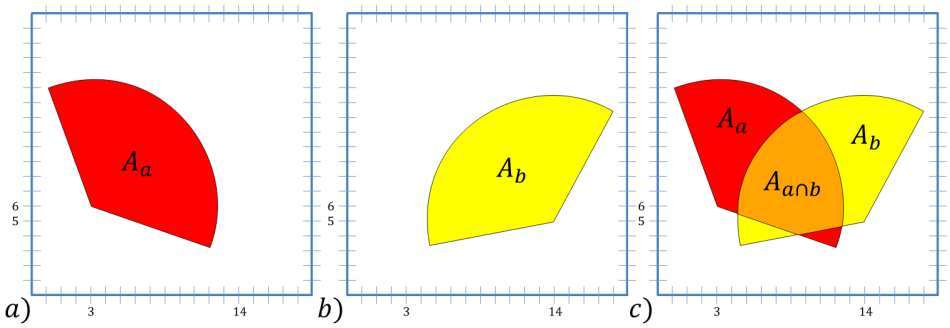


Figura 5: Exemplo de intersecção de áreas gerada. A área vermelha corresponde apenas a uma imagem acústica capturada pelo sonar, presente na Figura 5a) e na Figura 5 c). A área amarela corresponde apenas a segunda imagem acústica capturada pelo sonar, presente na Figura 5 b) e na Figura 5 c). A área laranja da Figura 5c) corresponde à intersecção das duas imagens.

3.1.2 Modelagem como Classificação Binária

Uma solução é modelar o problema de *matching* de imagens de FLS como um problema de classificação binária. As amostras são consideradas como par positivo quando representam o mesmo tipo de saída e negativo quando são diferentes entre si. É necessário anotar os pares de imagens com base em algum critério. O critério adotado, por sua vez pode ser baseado em *similaridade semântica*, *proximidade geográfica das capturas das cenas* ou *similaridade entre as imagens*.

Diferentes critérios irão gerar diferentes classes de pares positivos e negativos. Ao comparar a proximidade geométrica das capturas, por exemplo, uma imagem I_A só terá *matches* ou correspondências com imagens que foram capturadas naquela localidade. Em uma trajetória de SLAM, apenas lugares onde há *loop* que haverão *matches*. Portanto a maior parte dos pares de uma mesma imagem de entrada, podem ser considerados negativos, se compararmos a mesma imagem com todas as outras. Pois existem menos pares próximos a localidade de I_A do que pares distantes, resultando assim em um desequilíbrio entre casos positivos e negativos.

Um critério que pode ser utilizado é a similaridade semântica entre as imagens. Uma imagem de FLS com barcos e postes é similar a outra imagem de FLS com barcos e postes independente de terem sido capturados em diferentes localidades. O problema deste tipo de anotação, é que ela deve ser feita para cada uma das imagens manualmente por um especialista.

Ao adotar o critério de similaridade de imagens acústicas, apenas imagens muito parecidas visualmente são consideradas correspondentes. Enquanto que imagens que capturam os mesmos objetos, mas possuem diferenças de rotação relativa altos, não são consideradas correspondentes. Isto restringe correspondências à imagens capturadas com uma pequena diferença de tempo entre si.

Ao utilizar classificação binária, com as imagens disponíveis nos *datasets* utilizados neste trabalho, é necessário tratar o problema de desequilíbrio entre as classes. Isto não invalida o uso das técnicas de classificação, apenas adiciona um fator de complexidade que deve ser levado em consideração. Weiss, McCarthy e Zabar (2007) realizam um estudo sobre algumas das abordagens para tratar o desequilíbrio de classes. Técnicas de *sampling* e *cost-sensitive learning* são abordadas no trabalho. *Sampling* envolve alterar a quantidade de amostras de cada classe. *Downsampling* neste contexto é usar menos amostras das classes com maior número de amostras e *Upsampling* é replicar o número de classes com maior número de amostras a fim de atingir um equilíbrio entre o número de amostras disponíveis para cada classe. *Cost-sensitive learning* envolve a utilização de custos diferentes para cada situação em uma matriz de confusão. O trabalho descreve motivos pelos quais cada abordagem pode ser utilizada e conclui que não há um vencedor definitivo entre as três abordagens. Com o intuito de equilibrar a quantidade de escores possíveis, em um dos experimentos foi realizado uma subamostragem (ou *downsampling*) dos escores de treinamento e validação.

3.1.3 Modelagem baseada em Regressão de Similaridade

Modelos de regressão, assim como modelos de classificação são aproximadores de funções. Entretanto, enquanto que classificadores discretizam suas saídas em classes, modelos de regressão estimam valores contínuos como saída. Ao utilizar uma arquitetura de regressão, tem-se um modelo capaz de predizer uma função de similiaridade $S(I_A, I_B)$ para duas imagens de FLS I_A e I_B.

Como a medida definida na subseção 3.1.1 retorna um escalar, o modelo de regressão teria como saída a dimensão de um escalar. Este modelo se torna uma função de similaridade que assume valores entre um limite mínimo $O_{mín}$ e um limite máximo $O_{máx}$. O limite máximo representa o maior grau de similaridade entre as duas imagens, enquanto que o limite mínimo representa nenhuma similaridade entre as duas imagens. Ao adotar-se um critério $S_c(I_A, I_B)$, calculado a *priori*, é necessário calcular a diferença entre $S(I_A, I_B)$ e $S_c(I_A, I_B)$. Esta diferença será chamada de erro ϵ_{erro} e será calculada por uma norma L^1. A equação 3.1 representa um erro com uma norma L_1 (distância absoluta). Logo, o problema de encontrar $S(I_A, I_B)$ é uma minimização do erro ϵ_{erro}.

$$\epsilon_{erro} = \|S(I_A, I_B) - S_c(I_A, I_B)\|_1 \tag{3.1}$$

Assumindo o $O_{mínimo}$ como 0 e $O_{máximo}$ como 1, e utilizando o critério $S_c(I_A, I_B)$ de intersecção de áreas das imagens acústicas representado na Figura 5, a função de similaridade $S(I_A, I_B)$ retorna a similaridade das imagens I_A e I_B com uma margem de erro ϵ_{erro}. Desta forma, o grau de similaridade da função $S(I_A, I_B)$ é aproximado através de uma estratégia de otimização da medida $S_c(I_A, I_B)$.

Do ponto de vista de detecção de *loops* basta reconhecer corretamente um ambiente visitado, classificando o par (I_A,I_B) como correspondente ou não correspondente. Para realizar a transição de modelo de regressão para modelo de classificação, o problema de detecção de *loops* é modelado como encontrar uma função booleana $F(I_A, I_B)$ descrita na Equação 3.2. $F(I_A, I_B)$ depende de uma função escalar $S(I_A, I_B)$ que representa o grau de similaridade entre as imagens. α como o limiar de similaridade adotado para que as duas imagens acústicas sejam consideradas correspondentes. Quando α é aplicado no modelo, diz-se que ele é α_s. Quando α é aplicado no *ground truth* de validação, diz-se que ele é α_{gt}.

$$F(I_a, I_b) = \begin{cases} 0, \text{ para } S(I_A, I_B) < \alpha \\ 1, \text{ para } S(I_A, I_B) \geq \alpha \end{cases} \tag{3.2}$$

3.1.4 Modelagem baseada em Aprendizagem de Métrica

Aprendizagem de métrica é uma área relacionada a aprendizagem de máquina. Na computação, ela é normalmente utilizada em conjunto com características representadas através de vetores para os mais diversos problemas. Dentre eles: reconhecimento de faces(SCHROFF; KALENICHENKO; PHILBIN, 2015), sugestão de perfis em sites de relacionamento (MCFEE; LANCKRIET, 2010), *zero-shot learning* (SONG et al., 2016a), comparação fina de similaridade (WANG et al., 2014) entre imagens e recuperação de imagens (HOI; LIU; CHANG, 2010).

Métricas ou funções de distância são funções matemáticas que definem a distância entre pontos de um conjunto (KULIS, 2010). Este conceito possui propriedades que são interessantes para o problema de comparação de vetores. As propriedades que definem uma métrica $d : \mathbb{X} \times \mathbb{X} \to [0, \infty)$ são quatro:

A propriedade de *Não-Negatividade* representada pela Equação 3.3,

$$d(x, y) \geq 0, \tag{3.3}$$

define que toda métrica deve ser não-negativa.

A Equação 3.4 define a propriedade de *Identidade* de uma métrica,

$$d(x, y) = 0 \Leftrightarrow x = y, \tag{3.4}$$

ao garantir que se a distância entre dois pontos é zero, eles são idênticos. Isto significa que ao aplicar uma métrica para dois vetores iguais, o resultado deve ser zero. E também significa que se dois vetores possuem distância zero entre si, eles são iguais perante a métrica.

A Equação 3.5 expressa a propriedade de *Simetria*

$$d(x, y) = d(y, x), \tag{3.5}$$

definindo que toda métrica deve ser simétrica. Isto significa que a distância entre x e y é igual a distância entre y e x.

A propriedade descrita na Equação 3.6

$$d(x, z) \leq d(x, y) + d(y, z) \tag{3.6}$$

define a *Desigualdade Triangular*. Ela indica, entre outros aspectos, a transitividade entre as proximidades de um trio de pontos. Se x é próximo de y e y é proximo de z, logo x é próximo de z.

Como exemplo de distâncias existem: a absoluta; a distância euclideana; e para conjuntos positivos semi-definidos a Mahalanobis e etc.

Considerando a propriedade de simetria, toda métrica garante que a distância entre as características de uma imagem de FLS I_A e as caraterísticas de uma imagem de FLS I_B é $d(I_A, I_B) = d(I_B, I_A)$. Ao utilizar uma arquitetura de regressão ou classificação, a arquitetura é aproximada ao conjunto de dados de uma função a ser aprendida. Entretanto, isto não significa que a função será simétrica, mas será *aproximadamente* simétrica se o treinamento for adequado.

Kulis (2010) classifica o uso de técnicas de aprendizagem de métrica sobre vários aspectos: elas podem ser lineares ou não lineares; supervisionados, não-supervisionados ou semi-supervisionados. As técnicas que são abordadas neste trabalho envolvem o uso de uma função de distância simples e popular, que é a distância L_2. Esta função de distância é usada em conjunto com uma CNNs extratora de características específicas para o problema de comparação de cenas subaquáticas utilizando imageamento acústico.

3.1.4.1 *Triplets*

O conceito de *Triplets* é utilizado em Aprendizagem de Métrica em diversos contextos. Tais como: *Funções de Custo* que diminuem a distância aprendida em função da similaridade de um par e aumentam a distância aprendida em função da dissimilaridade de um par (WEINBERGER; SAUL, 2009); *Redes Triplets* que são redes neurais projetadas para explorar relações de similaridade e dissimilaridade nas suas etapas de treinamento e inferência (HOFFER; AILON, 2015)(WANG et al., 2014).

Weinberger e Saul (2009) já utilizava o termo para se referir a métodos de formulação do problema de aprendizagem de métrica como otimização convexa ou classificação de larga margem entre elementos. O *Large Margin Nearest Neighbor* - Vizinho Mais Próximo de Margem Larga (tradução livre) (LMNN) proposto pelos autores utiliza o termo *triplet* para relacionar a tupla de vizinhos-alvo (*target neighbors* - vizinhos da mesma classe em um *k-Nearest Neighbor* - k-Vizinhos Mais Próximos (kNN)) e impostores (elementos de classes diferentes).

Wang et al. (2014) propõe o uso de uma amostragem de *triplets* em um *dataset* designado para encontrar pares de amostras compostas por entradas de três elementos: um elemento âncora, um elemento da mesma classe ou similar ao âncora e um elemento de classe diferente ou dissimilar do elemento âncora. O objetivo do trabalho é a composição de uma rede para aprender ranking de imagens com base na similaridade fina entre elas. O trabalho atinge resultado superior aos modelos estado-da-arte que se baseavam em detectores de características clássicos.

Hoffer e Ailon (2015) propõe uma rede neural convolutiva para classificação que não aprende as classes diretamente, mas uma representação em um espaço métrico de dimensão 128. As comparações são feitas com uma rede convolutiva do tipo siamesa e os autores obtém melhores resultados nos conjuntos de dados: *Modified National Institute of Standards and Technology* database (MNIST)(LECUN et al., 1998), *Canadian Institute For Advanced Research* - 10 classes database (CIFAR10)(KRIZHEVSKY; HINTON, 2009), *Street View House Numbers* - *dataset* (SVHN)(NETZER et al., 2011) e STL-10 *dataset* (STL10)(COATES; NG; LEE, 2011).

3.1.5 Considerações sobre as modelagens

É possível então propor uma arquitetura treinada diretamente para classificação de cenas subaquáticas quanto à sua similaridade. Entretanto, ainda que considerando apenas um *ground truth* gerado pela medida proposta, qual seria o limiar de intersecção ótimo para realizar *matching* entre duas imagens de FLS? Se o limiar for muito alto, apenas imagens consecutivas serão correspondentes. Se o limiar for muito baixo, qualquer imagem com pouca intersecção de campos de visão pode ser considerada correspondente. Estritamente no aspecto de classificação binária: após definido o limiar, qual estratégia adotar para lidar com a diferença entre casos positivos e negativos? Quanto maior o limiar adotado, menos serão as amostras positivas. Quanto menor o limiar adotado, aumentam o número de pares negativos e diminuem os positivos.

A abordagem de modelar uma CNN de regressão parece uma alternativa mais interessante, pois adia-se o problema da escolha dos limiares para depois do treinamento. Entretanto, surge um problema quanto a maneira como o modelo é otimizado. Dependendo dos pares de imagens e rótulos escolhidos para treinamento, a rede pode ignorar

os valores de entrada e sofrer um sobreajuste em função de ser muito mais fácil da saída convergir para uma média global dos rótulos. Isto acontece quando o modelo não for suficientemente capaz de mapear as entradas para as saídas. Também ocorre em situações onde há um desequilíbrio muito grande das amostras para um determinado valor. Por exemplo, se 90% dos pares forem completamente não correspondentes, seu valor será $O_{mínimo}$ que é zero, enquanto que alguns poucos casos de intersecção se situaram entre $O_{mínimo}$ e $O_{máximo}$. Se em média os valores correspondentes forem $0,5$ (50% de intersecção), a média global tenderia para $0,05$ (escore de cinco centésimos). Portanto, sem efetuar uma seleção equilibrada das amostras, a rede pode facilmente convergir para um valor médio e ser insensível às entradas.

O importante destacar é que a abordagem de regressão não é melhor ou pior que a de classificação. Apenas surgem características diferentes no processo de treinamento que devem ser observadas. Se o modelo de regressão para o problema aqui tratado fosse perfeito em relação a medida proposta, ele por um lado seria mais útil que o classificador pois apresentaria não apenas uma relação de quais imagens são similares, mas sim o quão similares elas são. Por outro lado, caso o erro dele seja muito alto, a escolha de limiares α_s pode não ser genérica para qualquer *dataset*. E apesar da flexibilidade de escolha de limiares, própria necessidade de ter que escolher um limiar adequado pós treinamento pode ser enviesada pois ela afeta drasticamente o desempenho do modelo de regressão como classificador.

Quanto a anotação dos dados, tanto a aprendizagem de regressão quanto classificação geram $C_2^n = \binom{n}{2} = \frac{n!}{2!(n-2)!}$ pares de n imagens anotadas. Percorrer todas as amostras possíveis possui uma complexidade $O(n^2)$ em função de n imagens. Ao utilizar alguma forma de aprendizagem de métrica unária, a quantidade de amostras segue a proporção 1:1 entre imagem de entrada e rótulo. Ao utilizar de *Triplets*, por exemplo, a complexidade para processar todas as amostras cresce cubicamente com o número de imagens (WANG et al., 2014).

Tendo em vista as considerações feitas, inicialmente o trabalho concebe uma arquitetura de regressão de similaridade de cenas subaquáticas para ser avaliada como classificador. Em seguida, a proposta de uma estratégia que utilize aprendizagem de métrica com aprendizagem profunda para aprendizadem de características de imagens de FLS para comparação de cenas subaquáticas foi desenvolvida.

3.2 Arquitetura de Regressão

Este trabalho propõe uma nova rede aprendizagem profunda chamada *Sonar Matching Network* - Rede de Correspondência para Sonar (SMNet). Um dos motivos pelos quais se optou por desenvolver uma arquitetura específica, foi ter flexibilidade de criar

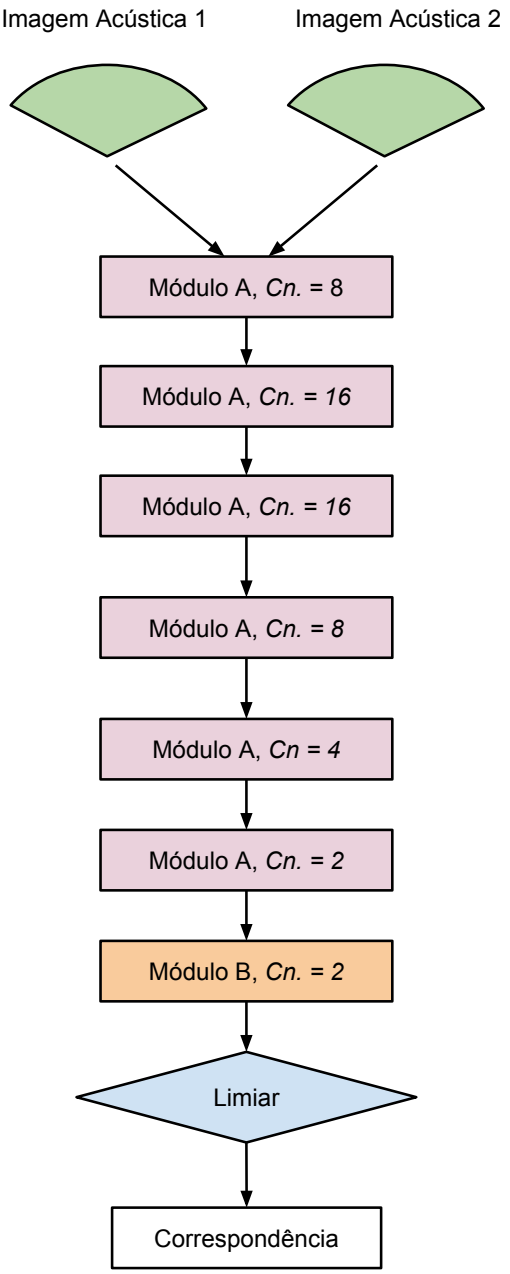


Figura 6: SMNet - Arquitetura Deep Learning: A arquitetura proposta é composta pelo encadeamento de dois tipos de módulos. O Módulo A aparece seis vezes na rede com uma variação no número de canais. O Módulo B aparece apenas no final da nossa rede e é responsável reduzir o volume de dados e calcular um Escore de Similaridade. Após a aplicação de um limiar neste Escore, pode-se classificar as imagens em Correspondentes e Não-Correspondentes. A rede é flexível para variações nos limiares utilizados após o treinamento.

uma arquitetura com poucos parâmetros com as técnicas que fossem consideradas mais adequadas para o problema de regressão de similaridade entre cenas subaquáticas. Isto possibilitou que modificações julgadas pertinentes para o problema fossem feitas sem a necessidade de manter um vínculo com algum modelo conhecido. O conhecimento adquirido durante o projeto desta arquitetura também ajudou a compreender quais aspectos de projeto de redes neurais eram mais relevantes para o problema de comparação de cenas.

A arquitetura convolutiva proposta para comparar as cenas recebe como entrada duas imagens acústicas. As imagens são obtidas por um FLS do modelo Blueview P900-130. Cada imagem acústica é uma imagem de 16bits de único canal. Diferentemente do trabalho de (VALDENEGRO-TORO, 2017), a SMNet processa a imagem por inteiro redimensionada. É importante ressaltar que na coleta de dados para treinamento e validação realizados por Valdenegro-Toro (2017), existe uma maior diversidade de estruturas do que os *datasets* coletados no Yacht Club e que foram selecionados para este trabalho. Os *datasets* selecionados para esta dissertação capturam apenas um pequeno conjunto de classes distintas, como barcos, postes e estruturas portuárias. Por este motivo, evitou-se utilizar neste trabalho a abordagem de classificação em *patches*.

A Figura 6 apresenta uma visão geral da SMNet. Internamente ela é organizada em módulos A e B cuja configuração de canais (Cn.) é variável. A saída da rede é um escore de similaridade entre as duas cenas. Com este escore, é aplicado um limiar α_s. Caso o valor seja superior ao limiar, a arquitetura considera as duas cenas como correspondentes.

Os módulos A possuem camadas com *kernels* de tamanho distintos, como pode ser visto na Figura 7. Muitas arquiteturas utilizam *kernels* de tamanho distintos em problemas de classificação (SZEGEDY et al., 2015)(HE et al., 2015), a rede MatchNet (HAN et al., 2015) que lida com outro tipo de *matching* também considera importante o tratamento de entradas com *kernels* em multi-escala. Para reduzir dimensionalmente a rede a um escalar é utilizado o Módulo B que pode ser visto na Figura 8.

A arquitetura também utiliza *Batch Normalization* ou Normalização de Lotes (BN)(IOFFE; SZEGEDY, 2015) para acelerar a convergência do treinamento e atenuar o problema da saturação dos gradientes. Outro aspecto interessante é a menor necessidade de preocupação com a inicialização dos pesos da rede e a BN já funcionar como uma espécie de regularizador de pesos. O problema da saturação dos gradientes ocorre com maior frequência em arquiteturas com muitas camadas de profundidade.

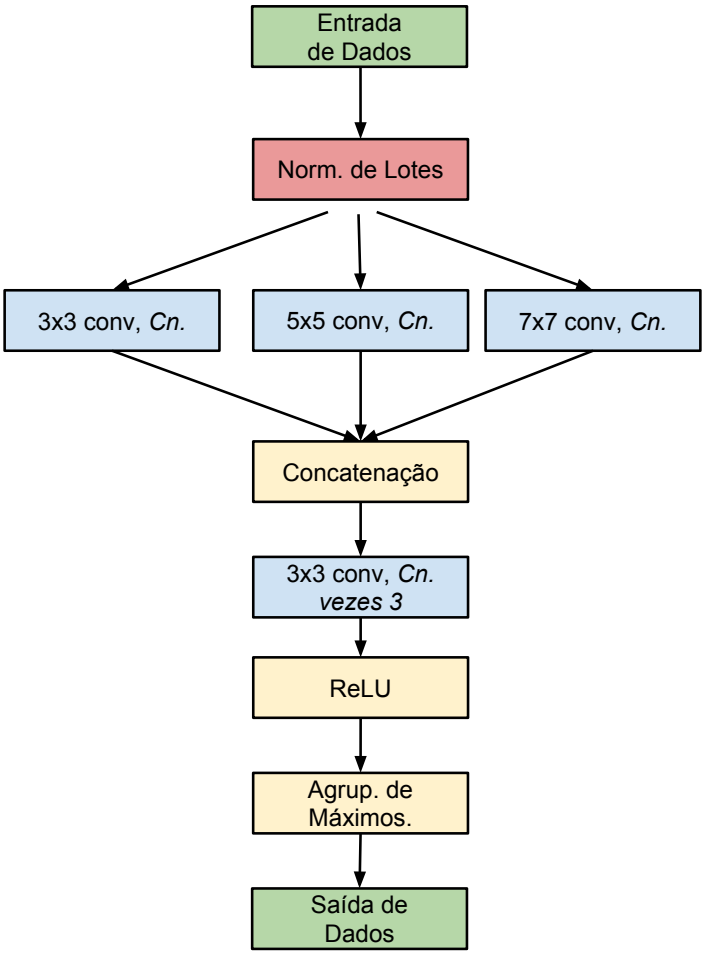


Figura 7: SMNet - Módulo A: Este módulo começa com uma Normalização de Lotes. Após o conjunto de entrada ser normalizado, são aplicadas convoluções com *kernels* de diferentes tamanhos. Os mapas de *features* obtidos após as diferentes convoluções são concatenados como diferentes canais. Outra convolução é feita em seguida, desta vez com três vezes o número de canais parametrizáveis. O módulo utiliza a função de ativação ReLU e realiza uma operação de *max pooling* com um *stride* de 2×2 para redução de metade do tamanho do volume de entrada.

3.2.1 Treinamento da Rede

Para realizar o treinamento da arquitetura proposta por este trabalho, é necessário que exista um *ground truth* ou em tradução livre: um conjunto de dados que represente os dados a serem preditos pela CNN. No problema endereçado por este trabalho, o *ground truth* é a medida proposta na Subseção 3.1.1. Com a posição física nas quais as imagens foram capturadas, é possível calcular a intersecção de duas áreas definidas por dois setores círculares. O raio do setor é definido pelo alcance do sonar utilizado no conjunto de dados. A integral da área comum aos campos de visão de I_A e I_B é uma medida que estima elementos em comum das duas cenas.

Foram selecionadas amostras distribuidas uniformemente entre as 10 faixas de valores, de 0 a 1,0 da medida de similaridade. Portanto, haviam 14.400 amostras entre 0 e 0,1; 14.400 entre 0,1 e 0,2; e assim sucessivamente. Pelo fato da maior parte dos pares ser

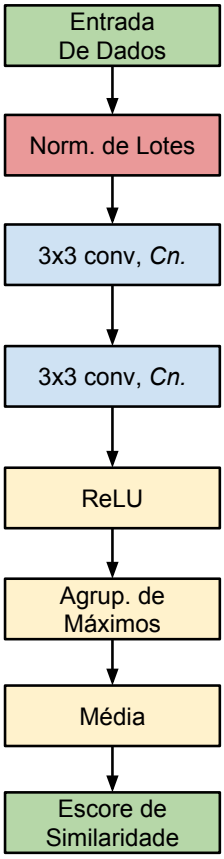


Figura 8: SMNet - Módulo B: Este módulo começa com uma Normalização de Lotes. Após o conjunto de entrada ser normalizado, são aplicadas duas convoluções consecutivas com *kernels* de tamanhos 3×3 . O módulo utiliza a função de ativação ReLU e realiza uma operação de *max pooling* com um *stride* de 2×2 para redução de metade do tamanho do volume de entrada. A média aritmética dos valores é calculada, gerando um número escalar, descrito como Escore de Similaridade.

0 e eles terem sido rejeitados, pode-se dizer que a técnica de subamostragem foi utilizada para lidar com desequilíbrio dos escores. Esta uniformidade de amostras foi garantida durante o processamento de cada mini-lote.

O método de otimização utilizado para reduzir o erro ϵ_{erro} foi o algoritmo ADAM (KINGMA; BA, 2014) baseado em *Gradient Descent* - Gradiente Descendente (GD) por ser mais flexível com o uso de dois *momentums* e apresentar melhores potenciais que outras estratégias de atualização de pesos com o ajuste fino de hiper parâmetros.

3.3 Aprendizagem de Métrica para imagens FLS

A aprendizagem de métrica combinada com aprendizagem profunda é usada atualmente em várias tarefas de comparação de imagens ópticas (HOFFER; AILON, 2015) (VO; HAYS, 2016) (SONG et al., 2016a). Uma das premissas fundamentais neste tipo de abordagem é aprender a relacionar objetos como similares e não-similares. A idéia prin-

cipal é a utilização de uma função de *loss* que aproxime elementos similares ou *Positivos* e distancie elementos não-similares ou *Negativos*. O erro é calculado a partir de um elemento âncora. E a cada iteração, escolhe-se um elemento âncora **A** com dois elementos: um positivo **P** e um negativo **N**. A função de *loss* então penaliza toda vez que a distância da âncora ao positivo, $D(A, P)$, for maior que a distância da âncora ao negativo $D(A, N)$. Considerando um vetor de *features* X e um vetor de *features* Y, a distância $D(X, Y)$ entre eles é expressa na Equação 3.7, que é uma métrica.

$$D(X, Y) = \|X - Y\|_2 \tag{3.7}$$

No trabalho de Vo e Hays (2016), diferentes abordagens de rede são discutidas e comparadas. Dentre elas estão: a abordagem siamesa, a classificadora, *Triplet* e híbrida entre *Triplet* e classificadora para o problema de geolocalização de imagens. A rede que eles utilizam para efetuar estas comparações são baseadas na AlexNet(KRIZHEVSKY; SUTSKEVER; HINTON, 2012). A arquitetura Triplet apresentou melhores resultados que a siamesa. E dentre elas, a função de *loss* que apresentou melhores resultados para este último tipo de arquitetura é descrita na Equação 3.8. A função de *loss* é a *Distance Based Logistic* (DBL) proposta no artigo para representar a distância entre os trios de elementos.

$$L(A, P, N) = log(1 + exp(D(A, P) - D(A, N))) \tag{3.8}$$

Na Figura 9 está representado o *pipeline* de treinamento proposto neste trabalho. Foi utilizada a *loss* definida na Equação 3.8 para treinamento, onde A é uma entrada âncora, P é uma entrada positiva em relação à ancora e N é uma entrada negativa em relação à âncora. Diferente do trabalho de geolocalização, foi utilizada uma variação da arquitetura Inception (SZEGEDY et al., 2017) para a aprendizagem. Uma imagem FLS é processada pela rede, que gera um vetor de *features*. Estas *features* são utilizadas para mapear relações entre as imagens.

Uma outra novidade é a utilização da intersecção de áreas como medida para pares positivos e negativos. Foram utilizados pares com mais de 60% de intersecção entre as áreas como positivos e exemplos com 10% ou menos de intersecção entre as áreas como negativos. Para cada imagem, foram selecionadas 40 amostras positivas e 40 amostras negativas. A ordem de apresentação de amostras é aleatória, mas toda vez que uma amostra positiva da imagem I_A é apresentada, também é apresentada uma amostra negativa. Esta estratégia é bastante simples e está muito longe de ser uma alternativa ótima. Song et al. (2016b) descrevem exemplos ainda mais complexos de funções de *loss* e estratégias de escolha de minilote para realizar um agrupamento correto das amostras.

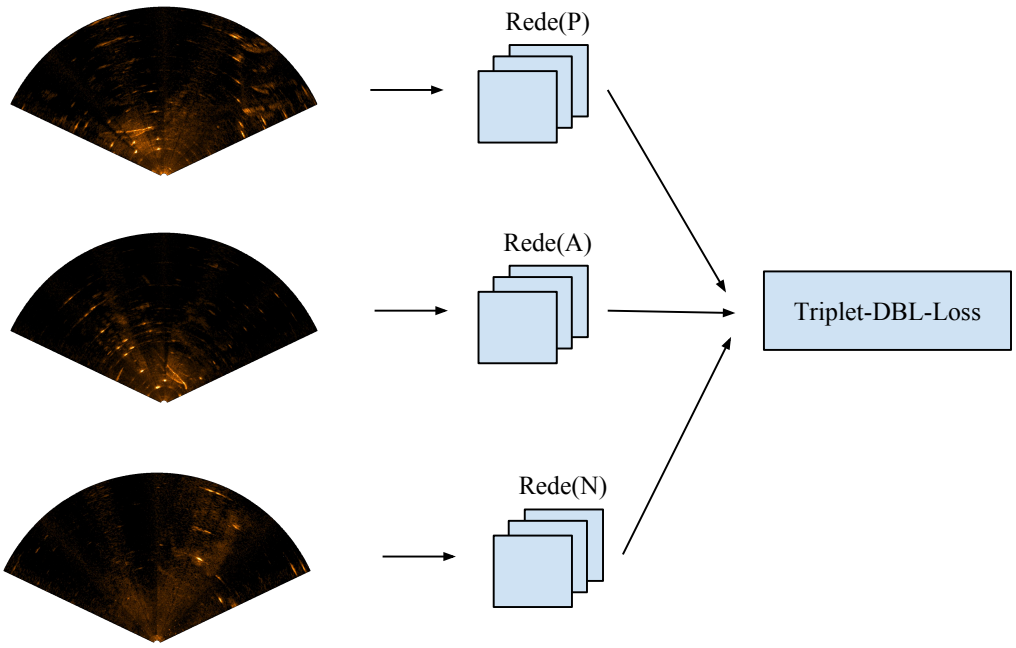


Figura 9: Neste *pipeline*, é utilizada uma CNN para mapear as imagens FLS em posições em vetor de 2048 elementos de ponto flutuante. A função de *loss* é aplicada sobre três instâncias da mesma arquitetura com pesos compartilhados. Esta CNN não aprende *features* para mapear a posição espacial de cada imagem. Ao invés disso, aprende a reconhecer *features* para estabelecer relações de distância n-dimensional entre as imagens. É utilizada a medida proposta nesta dissertação para definir quais entradas são positivas e quais são negativas.

3.4 *Datasets* ou Conjuntos de Dados

Um grande desafio em abordagens de aprendizagem de máquina tornar o modelo capaz de efetuar predições genéricas o suficiente. Existem diferentes razões para as quais um modelo não é genérico o suficiente, dentre elas: I) O modelo pode não ser capaz de associar corretamente as entradas e a saídas do fenômeno a ser aprendido. II) Os dados utilizados em treinamento podem não ser suficientemente genéricos para representar o problema. III) O modelo pode sofrer um sobreajuste para os dados de treinamento. De forma simplista, *overfitting* ou sobreajuste é um fenômeno que ocorre quando um modelo apresenta bons resultados em um *dataset*, mas se comporta de forma errônea em outros *datasets* do mesmo problema em que também foi modelado para efetuar predições. O sobreajuste é um problema conhecido na estatística, ocorrendo quando as amostras utilizadas para realizar inferência da população são tendenciosas. Em aprendizagem de máquina, diz-se então que o modelo não aprendeu corretamente o mapeamento de funções que deveria aprender, mas "memorizou"os dados.

Uma abordagem para avaliar possiveis sobreajustes em aprendizagem de máquina é a Validação Cruzada - *Cross-validation* (VC) que compreende diferentes técnicas de particionamento dos dados disponíveis para treinamento de um modelo (HAYKIN, 2007). Esta abordagem permite avaliar como um modelo se comporta com dados para os quais

ele ainda não foi calibrado dentro de um mesmo *dataset*.

Recentemente, Zhang et al. (2016) demonstra que o problema de sobreajustes é ainda mais complexo ao lidar com redes de aprendizagem profunda. Redes de aprendizagem profunda apresentam milhões de parâmetros, sendo capazes de se ajustar perfeitamente a rótulos aleatórios para imagens. Este comportamento levanta novas questões sobre quão capazes os modelos são de representar um fenômeno, ou o quanto eles são capazes de memorizar e mapear as entradas e saídas que são usadas para descrever o fenômeno.

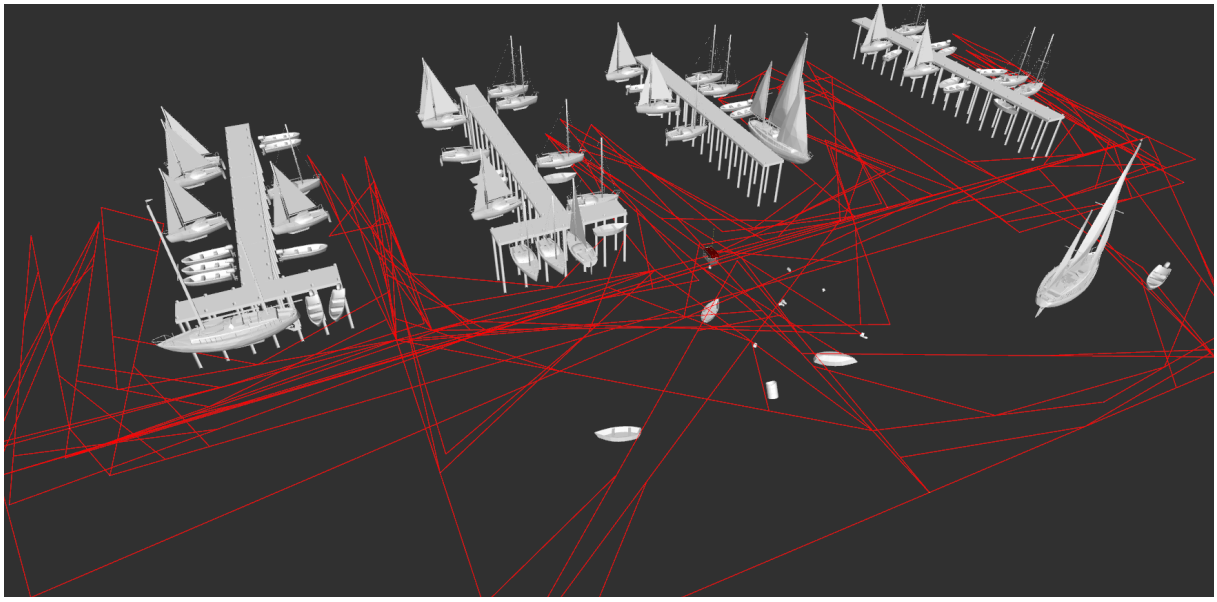
Treinar com um *dataset* e validar com um diferente não resolve o problema do *overfitting*, mas ajuda no processo de avaliação do modelo. Para visualizar como uma arquitetura efetua predições em ambientes desconhecidos (e.g. localizações diferentes) ou domínios diferentes do que foram inicialmente treinadas (e.g. cenas simulados contra dados reais) foram reunidos neste trabalho três *datasets* de ambientes distintos. É necessário notar que apesar do termo *dataset* ser utilizado tanto em robótica quanto aprendizagem de máquina, seus significados podem variar. Em robótica é comum vermos um conjunto de dados de sensores sem processamento para serem usados como objeto de pesquisa (STURM et al., 2012). Enquanto que em aprendizagem de máquina supervisionada, os *datasets* possuem anotações de saída do modelo que devem ser aprendidas (KRIZHEVSKY; HINTON, 2009). Por este motivo, ainda que tanto uma arquitetura de regressão e uma arquitetura de métrica usem o mesmo *dataset* em termos de robótica, ele pode ser construido de forma diferente em termos de aprendizagem supervisionada.

Para avaliar o quão genéricos são os modelos disponíveis, foram selecionados três *datasets*: Um Simulado (LONGARAY, 2017), um real de 2014 e outro real coletado em 2017 pelo grupo de pesquisa NAUTEC. Uma visão geral das trajetórias dos três pode ser vista na Figura 10.

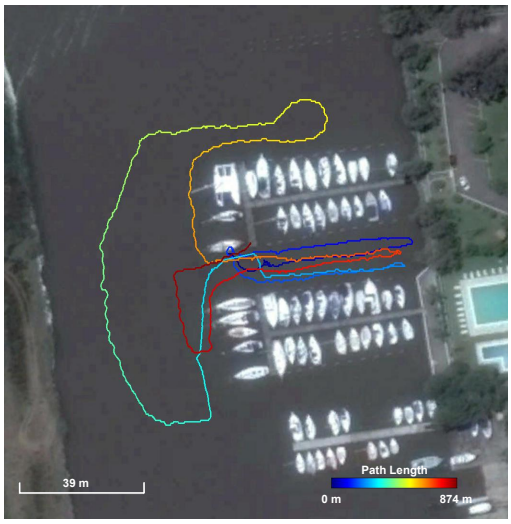
3.4.1 *Dataset* Simulado

Longaray (2017) desenvolveu uma rotina de coleta de dados sintéticos de FLS integrando diferentes interfaces de simuladores de robótica com um simulador de sonar em GPU. O renderizador(CERQUEIRA et al.,) foi calibrado para ter um ângulo de abertura igual ao do Blueview P900-130. O conjunto contém 10.997 imagens simuladas com leituras dos sensores de odometria em cada instante. As imagens capturam uma cena modelada pelo autor desenvolvida especificamente para a aplicação de exploração e inspeção de zonas portuárias. Foram modelados barcos e píeres para a cena 3D, como exibido na Figura 11.

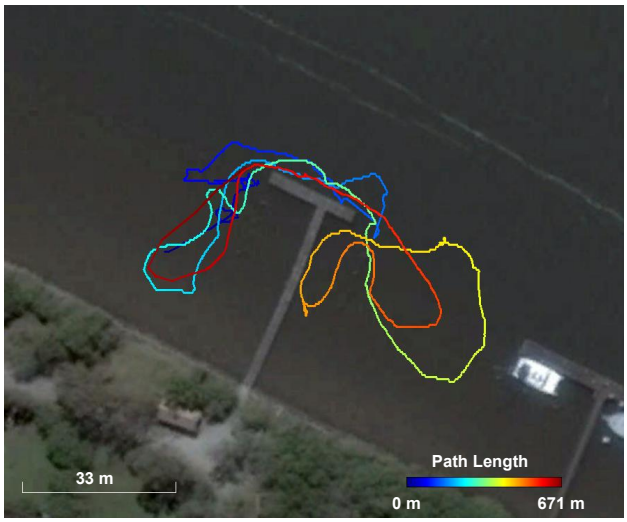
A Figura 12 estabelece um comparativo entre uma imagem simulada e uma imagem real. Conjuntos de dados sintéticos fornecem uma alternativa para pesquisadores



(a) Simulado



(b) ARACATI 2014



(c) ARACATI 2017

Figura 10: Trajetória dos três *datasets*: Simulado, ARACATI 2014 e ARACATI 2017: O *dataset* simulado foi confeccionado para conter elementos portuários comuns aos reais. Entre conjuntos de dados reais, o novo conjunto contém imagens capturadas com maior freqüência, explorando uma mesma cena por várias perspectivas diferentes.

interessados trabalhar com um fenômeno mas não dispõe dos sensores.

3.4.2 *Dataset* ARACATI 2014

Durante uma missão do ROV realizada pelo grupo de pesquisa em robótica NAUTEC, foram coletados dados de sonar e GPS junto ao robô. Este *dataset* é conhecido como ARACATI (SILVEIRA et al., 2015). Este dataset contém 10231 imagens de FLS obtidas no Yacht club. As imagens capturam barcos, trapiches e o fundo da região subaquática. As leituras dos sensores *Differential Global Positioning System* - Sistema de Posicionamento Global Diferencial (DGPS) e bússola possuíam uma freqüência de operação menor

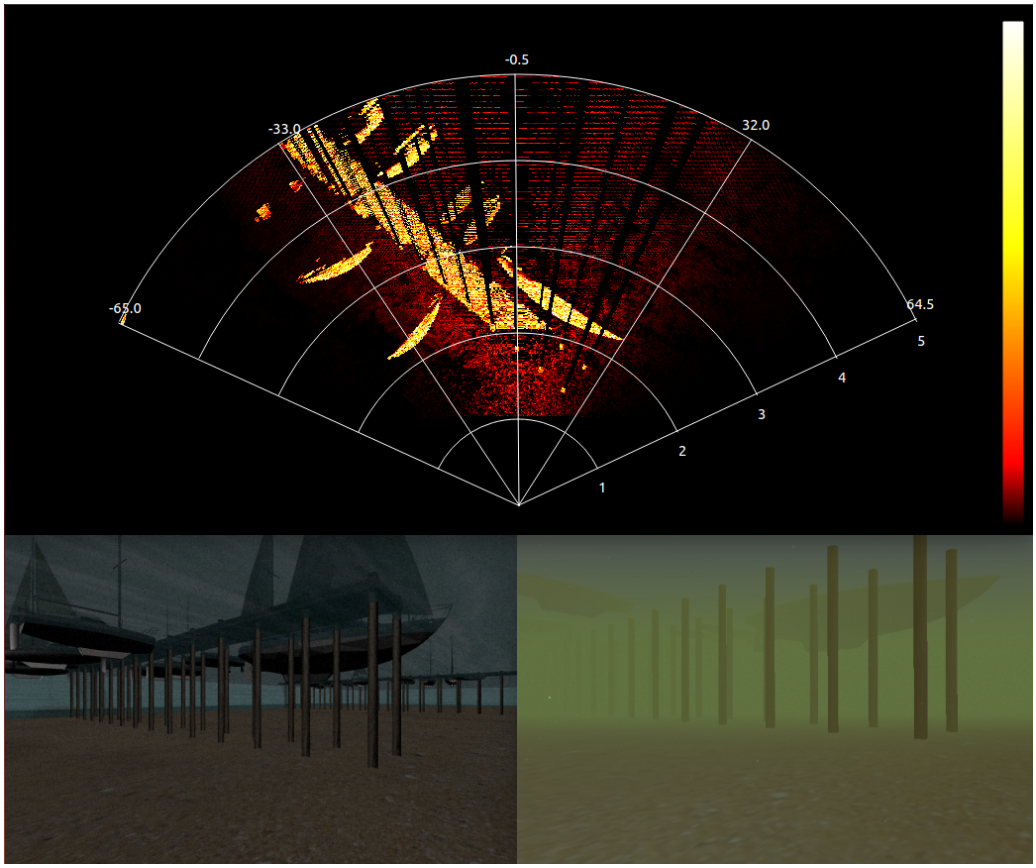
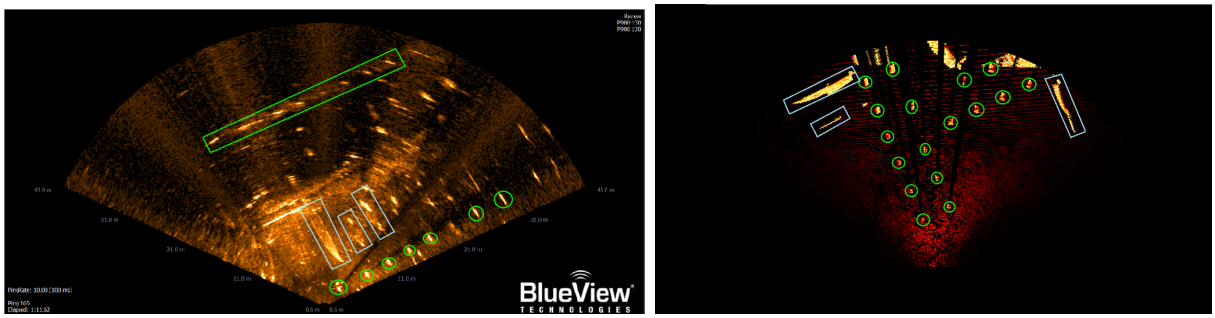


Figura 11: *Dataset* Simulado: Exemplo de cenário 3D construído em um simulador para representação da navegação do ROV usando um FLS. Nas imagens inferiores é mostrada uma simulação de uma câmera subaquática. Inferior à esquerda sem turbidez e simulando a turbidez à direita (LONGARAY, 2017).



(a) Imagem Real (b) Imagem Simulada

Figura 12: Comparação lado-a-lado imagem real e imagem simulada: Figura 12(a) mostra uma imagem de FLS capturada no Yacht Club, enquanto que a Figura 12(b) demonstra uma imagem renderizada. Retângulos brancos foram utilizados para destacar os barcos. Postes de madeira foram destacados com círculos. É interessante notar que a imagem simulada também apresenta sombra acústica (Adaptado de Longaray (2017)).

que a freqüência do FLS. Isto significa que entre um número expressivo de imagens não existiriam leituras de sensores para calcular a diferença de rotação e translação do robô entre as capturas. Reduzindo assim o número de imagens que poderiam ser anotadas. Em alguns trabalhos do grupo, foram utilizadas apenas 3674 destas imagens em função das diferenças entre as freqüencias de operação. Para aumentar o número de imagens anotadas, foi empregado o uso de técnicas de interpolação com *splines* para translação e interpo-

lação linear para rotação, potencializando a criação de um conjunto de 10231 imagens anotadas com posição e rotação. A trajetetória do ARACATI 2014 pode ser observada Figura 10(b). Na Figura 13 é possível observar o modelo de ROV e FLS utilizados.

Figura 13: O robô subaquático utilizado para coleta de dados é o Seabotix LBV 300-5 acoplado com o FLS Blueview P900-130. Este sonar possui um amplo campo de visão, atingindo 130 graus de abertura

3.4.3 *Dataset* ARACATI 2017

Este dataset contém 25499 imagens de FLS obtidas no Yacht club. As imagens capturam barcos, trapiches e o fundo da região subaquática. As leituras dos sensores DGPS e bússola possuíam uma freqüência de operação menor que a freqüência do FLS. Isto significa que entre um número expressivo de imagens não existiriam leituras de sensores para calcular a diferença de rotação e translação do robô entre as capturas. Para aumentar o número de imagens anotadas, foi empregado o uso de técnicas de interpolação com *splines* para translação e interpolação linear para rotação. O conjunto de dados obtido na missão de 2017 situa-se em outra região física do Yacht club, como pode ser observado na Figura 10(c).

4 Resultados Experimentais

Neste capítulo encontram-se: A Seção 4.1 descreve as métricas de avaliação aplicadas no contexto de classificação binária; Seção 4.2 traz as avaliações realizadas sobre a arquitetura de regressão; A Seção 4.3 comenta um trabalho que foi utilizado como comparação para a arquitetura de métrica proposta.

4.1 Métricas para Avaliação

Para avaliar o desempenho da rede quanto à sua capacidade de detecção de pontos de fechamento de loop, foram utilizadas métricas de classificação binária.

4.1.1 Curva Precisão e Revocação

As curvas de *Precision and Recall* - Precisão e Revocação (PR) relacionam a transição entre o valor máximo de Precisão e o valor máximo de Revocação obtido por um classificador em condições de validação ao variar seus limiares de decisão. A Equação 4.1 representa a métrica de Precisão. VP representa os verdadeiros positivos. Falso Positivo (FP) representa os falsos positivos.

$$Precisão = \frac{VP}{VP + FP} \tag{4.1}$$

Uma grande preocupação em um sistema de reconhecimento de cenas é reconhecer correspondências verdadeiras com a Precisão próxima de 100%. Isto se deve ao fato de que quando sistemas de reconhecimento de lugares estão integrados a um sistemas de SLAM, eles são utilizados para corrigir a estimativa de trajetória de um robô móvel. Portanto, caso ocorra um *Falso Positivo*, o robô vai atualizar sua posição no seu sistema de forma equivocada. Isto comprometeria o funcionamento do sistema de SLAM. Na revisão bibliográfica de Lowry et al. (2016), os autores destacam que esta preocupação se tornou menor com a utilização de métodos capazes de corrigir *Falsos Positivos*.

A Equação 4.2 representa a métrica de Revocação. VP representa os verdadeiros positivos. Falso Negativo (FN) representa os falsos neativos. Ela também é conhecida como sensibilidade, pois representa o quanto um classificador consegue identificar casos verdadeiros. Em um cenário de reconhecimento de locais ou fechamento de *loops*, uma revocação alta representa que o modelo consegue detectar mais imagens de cenas que são

de fato correspondentes.

$$Revocação = \frac{VP}{VP + FN} \tag{4.2}$$

4.1.2 Curva Característica de Operação do Receptor

As curvas de *Receiver Operating Characteristic* - Característica de Operação do Receptor (COR) são comumente usadas nas Ciências Biológicas e Médicas (HANLEY; MCNEIL, 1982). Esta curva possui como eixos a Revocação e 1-Especificidade. A curva COR representa o comportamento de um classificador quanto à sua melhor revocação e à sua melhor especificidade. Os pontos são obtidos ao se variar o limiar de decisão entre o limiar que resulta na melhor revocação e o limiar que resulta na melhor especificidade. A área sobre a curva é utilizada como indicador da qualidade do classificador. Uma área de 0.5 representa um classificador que não é melhor do que um classificador aleatório. Uma área igual a 1.0 representaria um classificador perfeito para o conjunto de dados amostrais.

A Equação 4.3 representa a métrica de Especificidade. Verdadeiro Negativo (VN) representa os verdadeiros negativos. FP representa os falsos positivos. É uma taxa de acertos para predições negativas.

$$Especificidade = \frac{VN}{VN + FP} \tag{4.3}$$

4.1.3 Coeficiente de Correlação de Matthews

Em problemas de classificação binária, o MCC calcula o desempenho do classificador equilibrando ambas as classes. Isto é útil para situações em que a distribuição de elementos é muito desproporcional. Se ao algoritmo de comparação de cenas subaquáticos for utilizado no contexto de fechamento de *loop*, espera-se que em um cenário que o robô visite cenas pela primeira vez, grande parte das comparações resultem em Falso. Isto tornaria o conjunto de amostras enviesado.

$$\text{MCC} = \frac{VP \times VN - FP \times FN}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}} \tag{4.4}$$

A Equação 4.4, descreve o MCC. O resultado do MCC pode ser qualquer valor do intervalo $[-1, 1]$. O valor 1 representa uma predição perfeita em ambas as classes, enquanto valor -1 representa uma predição invertida (e.g. todos os falsos são considerados verdadeiros e vice-versa). O valor 0 representa que o classificador não se comporta melhor que um classificador aleatório.

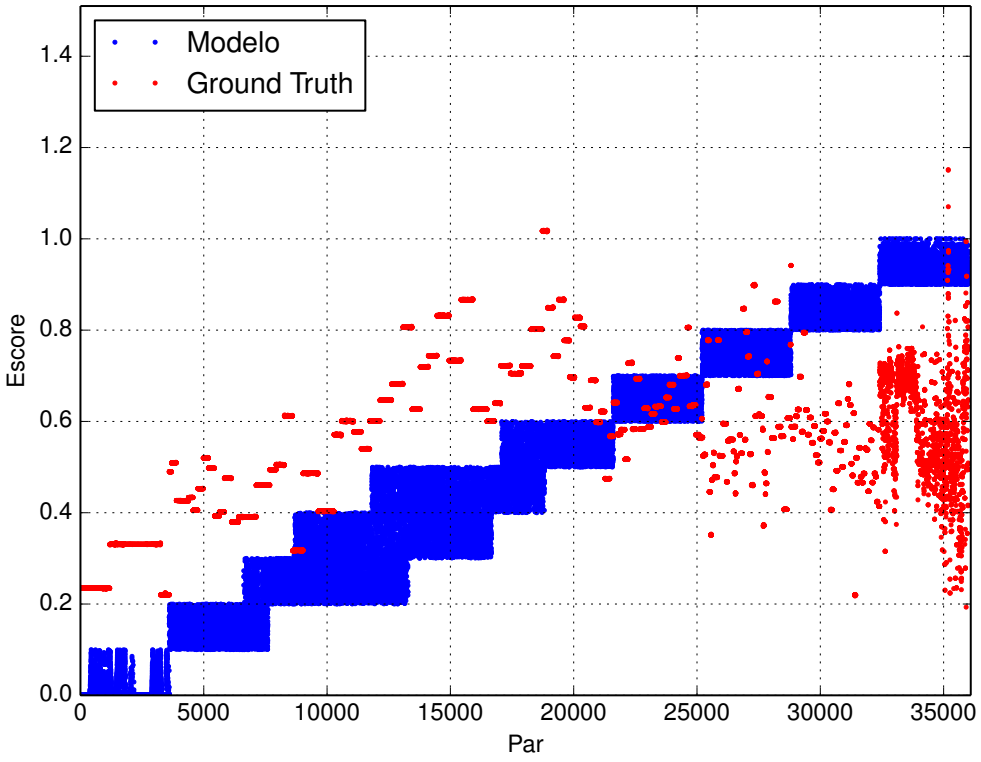


Figura 14: Curva de Regressão: esta curva é a saída do modelo de regressão para imagens Polares comparada diretamente com os pares do *ground truth*.

4.2 Avaliação da Arquitetura de Regressão

A saída direta da rede de regressão para o conjunto de pares de validação pode ser vista na Figura 14. É possível observar que a saída da arquitetura proposta (em vermelho) não consegue predizer perfeitamente o valor de cada (par em azul). Entretanto, com o uso de limiarização, a rede ainda pode ser aplicada para classificação de cenas como correspondentes.

Dado um par de imagens acústicas, considera-se que valores acima de α_{gt} de interseção encontrados pelo *ground truth* do conjunto de dados representam a mesma cena. Ou seja, assume-se que nestes casos há uma detecção de ciclo, pois se tratam de cenas relacionadas de acordo com a orientação do sonar e a posição GPS.

4.2.1 Curva de Regressão

Do conjunto de dados obtidos do ARACATI 2014, foram utilizadas 3674 imagens de FLS coletadas pelo ROV. Essas imagens foram amostradas em pares, e para cada par foi calculado o valor da medida de *ground truth* proposto. Para o conjunto de treinamento existiam 144.000 pares de imagens. A rede foi validada com um conjunto de 36.000 pares

de imagens não apresentados durante o treinamento. Os pares foram selecionados de acordo com intervalos de 0.1 em 0.1 de escores possíveis. Caracterizando 10 possíveis intervalos para cada faixa de valor possível da medida de intersecção de áreas. Tanto o conjunto de treinamento quanto o conjunto de validação possuem a mesma proporção de pares para cada uma das 10 faixas de valores possíveis. Na Figura 14 é possível observar a saída direta do modelo de regressão para os pares selecionados. Obviamente, a arquitetura possui dificuldades para associar corretamente o escore ao par de imagens. Entretanto, ainda é possível utilizá-la como um classificador binário de cenas utilizando de limiares escalares.

4.2.2 Curva Precisão e Revocação da Arquitetura de Regressão

Na curva PR, efetuou-se uma variação do limiar α_{gt}, que é o limiar do *ground truth*. Para diferentes valores de α_{gt}, buscou-se variar também os valores de α_s para obter diferentes valores de *Precisão*. A variação do limiar α_s também afeta os valores da métrica de *Revocação*. A curva PR mostra os pares das métricas (revocação,precisão) para todos os valores de α_s situados no intervalo $[\alpha_{mínimo}, \alpha_{máximo}]$, discretizados em passos (e.g. 0,001), para um α_{gt} fixo. $\alpha_{mínimo} = 0.1932$, que é o menor valor de saída do modelo. $\alpha_{máximo} = 1.1516$, que é o maior valor de saída do modelo.

É possível verificar que o classificador possui uma Precisão e Revocação maiores para valores entre 10% e 30% de intersecção na Figuras 15. A partir do valor de 40% de intersecção de cenas, o desempenho do classificador cai consideravelmente. Chegando a valores de Revocação quase nulos para quando a Precisão se aproxima de 1. Uma interpretação é que a rede classifica mais casos corretamente por possuir menor restrição de quantos pares possuem de fato uma alta similaridade. Predizendo que tanto casos de pares com pouca similaridade, quanto casos de pares com muita similaridade são correspondentes.

4.2.3 Curva Característica de Operação do Receptor da Arquitetura de Regressão

Na curva COR, efetuou-se uma variação do limiar α_{gt}, que é o limiar do *ground truth*. Com a variação do α_{gt}, buscou-se o melhor $\alpha_{especificidade}$ para aquele α_{gt}. O melhor $\alpha_{especificidade}$ neste caso representa o $\alpha_{especificidade}$ que retorna a maior especificidade possível. O outro limiar a ser buscado é o $\alpha_{revocação}$, que representa a maior revocação encontrada pelo modelo. A curva COR mostra os pares das métricas (1-especificidade, revocação) para todos os valores de α_s situados no intervalo $[\alpha_{mínimo}, \alpha_{máximo}]$, discretizados em passos (e.g. 0,001), para um α_{gt} definido. $\alpha_{mínimo} = 0.1932$, que é o menor valor de saída do modelo. $\alpha_{máximo} = 1.1516$, que é o maior valor de saída do modelo.

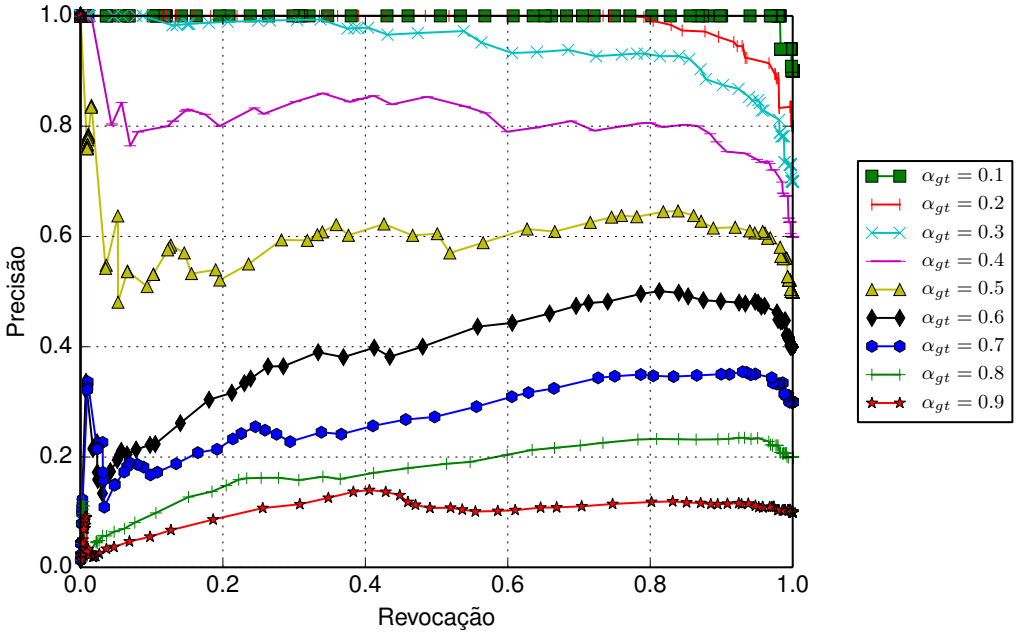


Figura 15: Curva de Precisão e Revocação: esta curva foi gerada para 10 valores possíveis do hiperparâmetro α_{gt} que determina o quanto de intersecção no mínimo é considerado como uma correspondência para o *ground truth*. A curva representa a variação de α_s, que é o limiar aplicado na rede, para o intervalo $[\alpha_{precisão}, \alpha_{revocação}]$

Na Figura 16 a área sobre a curva é consideravelmente alta para valores baixos de α_{gt}. Sugerindo que o classificador seja genérico tanto para casos negativos quanto positivos. Entretanto, diferente da curva de Precisão e Revocação, os limiares mais altos ($\alpha_{gt} \geq 0{,}5$) não denunciam tanto a inaptidão do classificador pois a área sob a curva é $\approx 0{,}5$.

4.2.4 Comparação da Arquitetura de Regressão com Grafos de Descrição Topológica

Machado, Drews-Jr e Botelho (2016) propuseram o método GDT para detecção de *loops* usando apenas FLS. Usando os parâmetros sugeridos pelo autor na Tabela 1, foi realizada uma comparação entre o GDT e a arquitetura de regressão sob o paradigma de classificação binária. Todos os testes foram realizados com o *ARACATI* 2014 com pares não vistos pela arquitetura. Os métodos tiveram seus parâmetros otimizados para maximizar o MCC e para maximizar a Precisão. O método proposto possui o melhor limiar α_s para ambas as métricas. Na Tabela 2 há a comparação de ambos os métodos otimizados para obter o melhor MCC. Neste caso, o GDT obteve valores mais baixos para

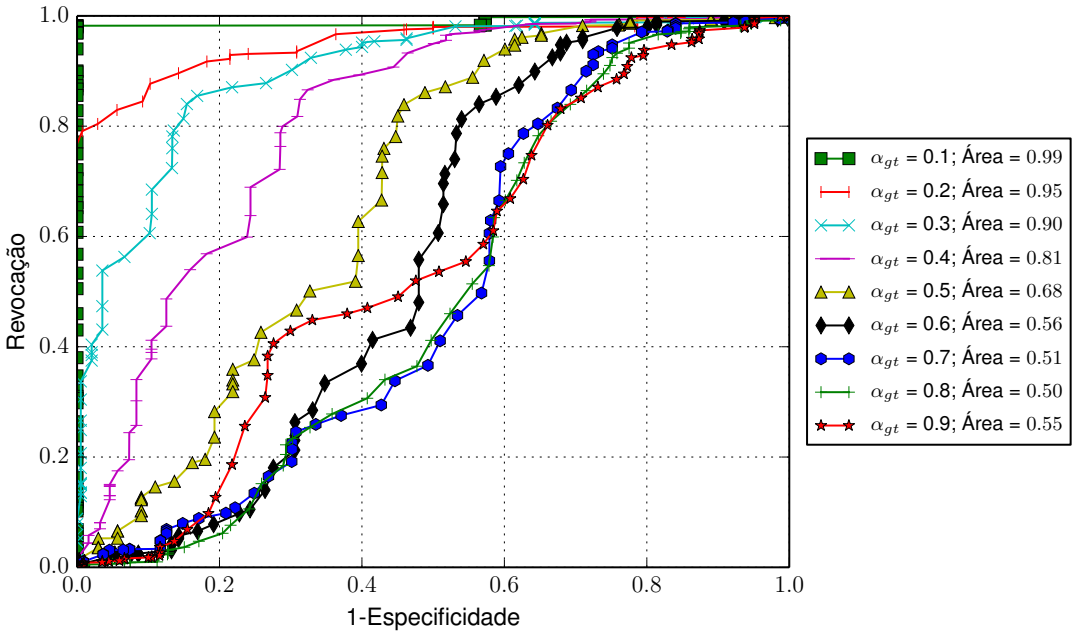


Figura 16: Curva de Característica de Operação do Receptor: esta curva foi gerada para 10 valores possíveis do hiperparâmetro α_{gt} que determina o quanto de intersecção no mínimo é considerado como correspondência para o *ground truth*. A curva representa a variação de α_s, que é o limiar aplicado na rede, para o intervalo $[\alpha_{revocação}, \alpha_{especificidade}]$

outras métricas também. Na Tabela 3 é possível perceber que ambos os métodos atingem 100% de precisão. Entretanto, o método proposto possuiu melhores resultados segundo outras métricas utilizadas para avaliar classificadores.

Tabela 1: Parâmetros utilizados pelo GDT.

Parâmetro	**Valor**
π_{fim}	0,6
$\pi_{recursivo}$	0,98
$dist$	4 pixels
mín. de pixels/Segmento	20 pixels
máx. de pixels/Segmento	12000 pixels
r	300 pixels
$\rho_{similar}$	4,0

Tabela 2: Comparativo do método de correspondência proposto com o GDT otimizado para MCC

	Método proposto	GDT
Acurácia	**98,41**%	50,57%
Sensibilidade	**98,24**%	45,83%
Eficiência	**99,12**%	69,51%
Predição Positiva	**100**%	98,37%
Predição Negativa	**86,35** %	16,05%
MCC	**0,92**	0,23

Tabela 3: Comparativo do método de correspondência proposto com o GDT otimizado para Precisão

	Método proposto	GDT
Acurácia	**98,41**%	13,27%
Sensibilidade	**98,24**%	2,47%
Eficiência	**99,12**%	51,15%
Predição Positiva	**100**%	**100**%
Predição Negativa	**86,35** %	10,23%
MCC	**0,92**	0,049

4.3 Avaliação da Arquitetura de Aprendizagem de Métrica

O objetivo desta arquitetura é comparar duas cenas subaquáticas através de suas respectivas imagens FLS, encontrando as melhores características para a recuperação de imagens FLS. Para avaliar a arquitetura de métrica proposta, a arquitetura foi treinada duas vezes. O primeiro modelo foi treinado apenas com dados de simulação e foi denominado *Metric Sim*. O segundo modelo foi treinado com dados reais do *dataset* ARACATI 2017 e foi definido como *Metric 2017*. **Todos modelos desta seção foram validados no conjunto de dados de 2014**. Portanto, os modelos *Sim* são treinados em simulação e validados com dados reais do Aracati 2014. Enquanto que os modelos *2017* foram treinados com dados reais de 2017 e validados em um novo local.

O trabalho relacionado de Li et al. (2016) é o mais próximo da proposta desta arquitetura. O trabalho dos autores utiliza múltiplos conjuntos de dados obtidos do casco do *SS Curtis*, mas nenhum deles foi tornado de domínio público. Nem as suas HDF estavam disponíveis na data atual. Por este motivo, foi implementada uma versão do trabalho HDF para fins de comparação. Observa-se o HDF foi um passo além no problema de localização ao utilizar a sua *imagem local* como entrada para um filtro de Kalman. Nesta dissertação, há o interesse apenas na avaliação de correspondência efetuada para recuperação de imagens que foi proposta no trabalho de Li *et al*, cujo *pipeline* de consulta de imagens baseada em conteúdo pode ser visto na Figura 17.

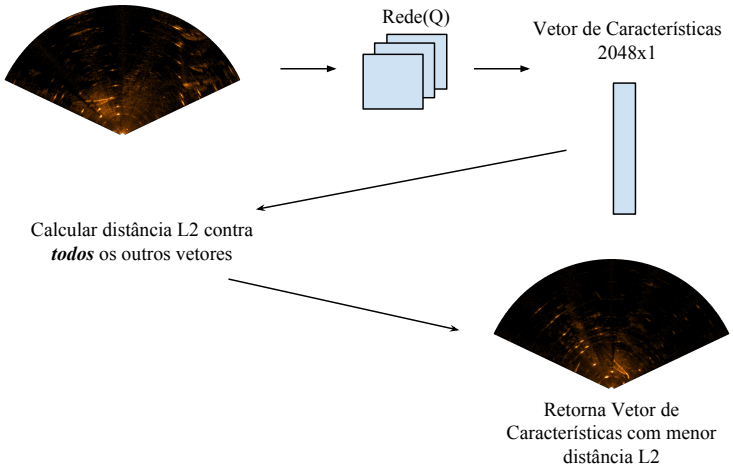


Figura 17: Neste *pipeline*, cada imagem de sonar é representada por um vetor de características. Os vetores são comparados entre si, e os pares com menor distância são considerados correspondentes. Desta forma é possível efetuar uma busca baseada em similaridades de características a partir de uma imagem Q processada pela rede.

4.3.1 Implementação da Arquitetura HDF

No trabalho de Li et al. (2016), a redução dimensional é feita com uma CNN para obtenção de HDFs. A CNN é treinada para realizar uma regressão da posição (x, y, z). Tanto o *dataset* quanto o código-fonte não estão disponíveis atualmente. Por este motivo, foi implementada uma versão do método proposto pelos autores com pequenas diferenças para fins de comparação. Os autores avançam ainda mais no problema de localização ao utilizar as *features* como entrada para um filtro de Kalman(SORENSON, 1985). Também são feitos ajustes com relação a incerteza das predições da rede e modificações específicas da arquitetura Inception para o sistema proposto pelos autores. Nesta dissertação, estamos apenas interessados na avaliação original de *matching* realizada da HDF para recuperação de imagens acústicas ao utilizar uma variação da arquitetura Inception treinada para predizer uma posição a partir de uma imagem acústica.

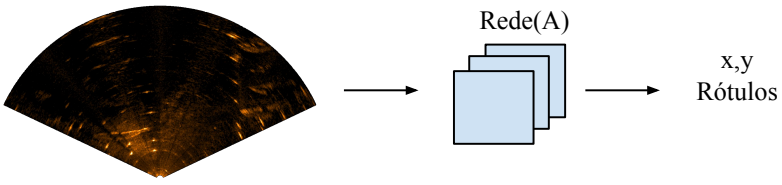


Figura 18: Neste *pipeline*, é utilizada uma CNN para mapear as imagens FLS em posições *x,y,z*. No conjunto de treinamento utilizado nesta dissertação, a distância em z é desprezível. Portanto, a rede só prediz valores nas demais dimensões espaciais.

O *pipeline* geral é mostrado na Figura 18. É importante notar que este trabalho utiliza *features* para o problema de localização cujo objeto de estudo é a inspeção de cascos de navio, mais especificamente o SS Curtis. Foram gerados múltiplos *datasets* do

mesmo casco em anos diferentes. Considerou-se que no trabalho dos autores que utilizam as HDF, estas *features* deveriam ser robustas para atender as variações das imagens do casco coletadas ao longo dos anos. O ambiente físico é o mesmo tanto para os conjuntos de dados treinamento quanto validação.

4.3.2 Convergência de Treinamento para HDF

A rede responsável pela estimativa de posição foi treinada em duas instâncias diferentes: a primeira com os dados do simulador e a segunda com dados reais de sonar. É possível verificar através do gráfico na Figura 19 que a rede convergiu muito bem para os dados de treinamento.

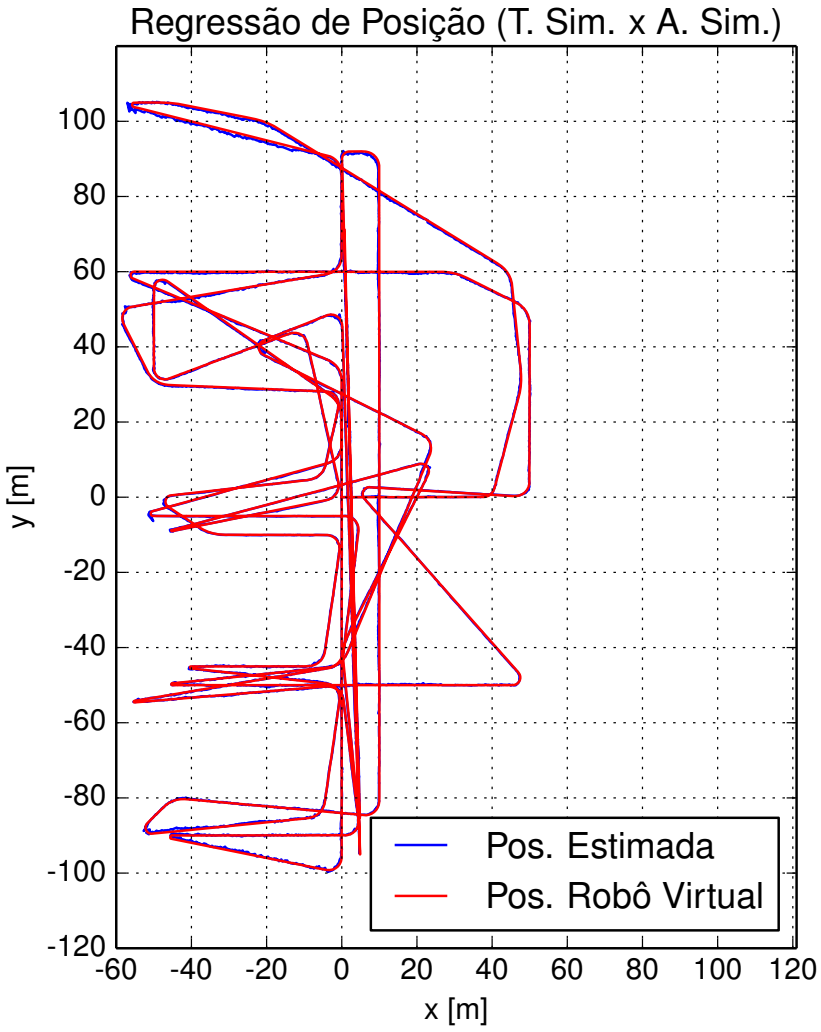


Figura 19: Predição em treinamento da implementação HDF: versão modificada da PoseNet para estimar posição (x, y) a partir de uma imagem de FLS treinada em ambiente de simulação. Este gráfico demonstra a convergência da estimativa de posição da rede em relação a saída desejada que é a posição do robô no *UWSIM*. É possível observar que a rede memoriza a posição em que cada imagem foi capturada.

Durante o treinamento com dados reais de sonar, observa-se na Figura 20 que a rede convergiu de forma suave em relação a posição coletada pelo GPS. Este erro de

treinamento é extremamente baixo para a complexidade do problema tratado. É correto afirmar que a arquitetura utilizada foi treinada de forma adequada.

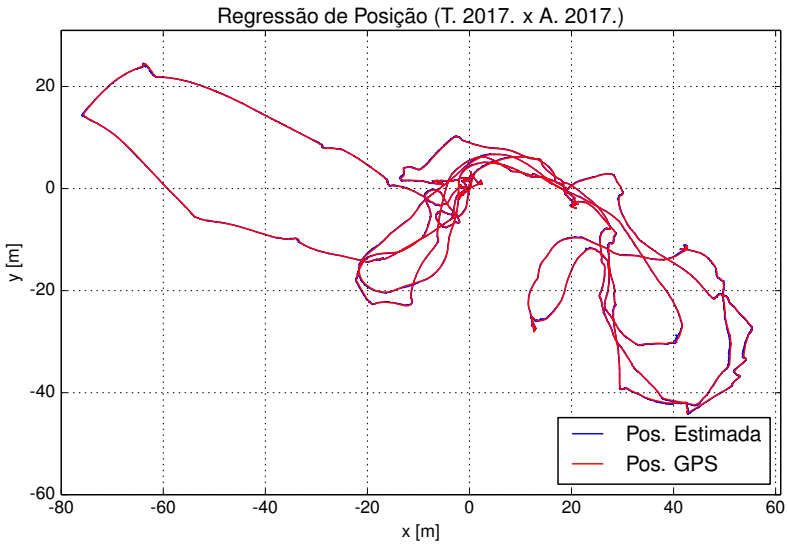


Figura 20: Predição em treinamento da implementação HDF: versão modificada da PoseNet para estimar posição (x, y) a partir de uma imagem de FLS treinada no *dataset* ARACATI de 2017. Este gráfico demonstra a convergência da estimativa de posição da rede em relação a saída desejada que é a posição do GPS. É possível observar que a rede memoriza a posição em que cada imagem foi capturada.

4.3.3 Comparação entre Aprendizagem de Métrica e HDF para *Matching* de Imagens Acústicas

A avaliação consiste em comparar cada vetor de características extraídos da imagem de FLS com todas as outras usando distância L^2. A imagem com menor distância é retornada da busca. A correspondência é considerada *Verdadeira* quando as duas imagens do par possuem uma intersecção de campos de visão superior a um determinado limiar de intersecção. No trabalho que foi usado como referência para implementação do HDF, foi definido um limiar de 50% de intersecção. Neste experimento, entretanto, os limiares de intersecção da área dos campos de visão foram calculados de forma variável.

Na Figura 21 é mostrada a acurácia de cada método utilizando *datasets* de treinamento e validação diferentes. Observa-se que as instâncias da arquitetura que foram treinadas com dados reais, HDF 2017 e *Metric* 2017, obtiveram melhores resultados que suas versões que utilizaram dados simulados. Isto acontece porque os conjuntos de dados reais de 2017 são mais parecidos com os conjuntos de 2014, do que os dados gerados em simulação. Ainda que as imagens sintéticas se pareçam com as reais, existem diferenças da escala do mapa de cor; padrão de ruído; resolução; intensidade de pixel e etc. Os re-

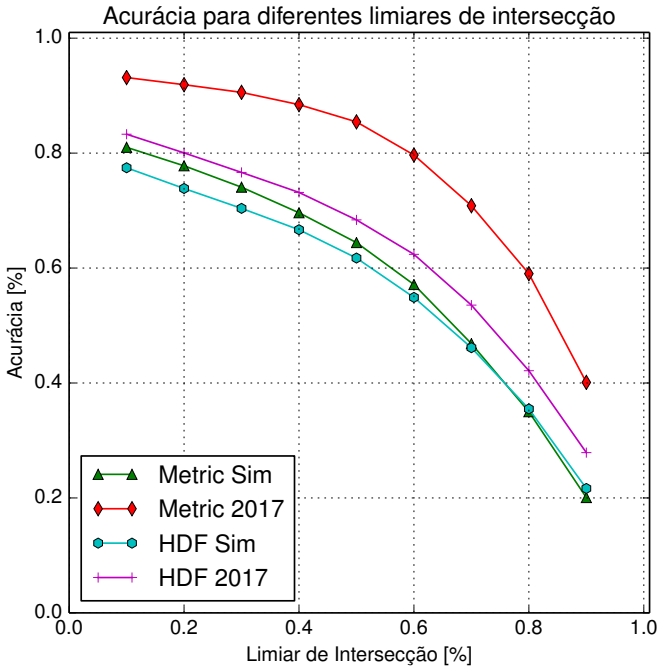


Figura 21: Acurácia para diferentes níveis de intersecções: a mesma arquitetura é treinada de quatro formas distintas. HDF Sim é treinada para regressão de Pose usando conjuntos de dados simulados e é validada com os dados de 2014. HDF 2017 é treinada para regressão de Pose usando conjuntos de dados reais do *dataset* de 2017 e validada com os dados de 2014. *Metric* Sim utiliza de aprendizagem de métrica com dados simulados e é validada com dados reais de 2014. *Metric* 2017 usa a abordagem de aprendizagem de métrica com dados de 2017 e é validada com os dados coletados em 2014.

sultados para dados simulados não obtiveram melhoras significativas entre as abordagens de *Metric* e HDF.

Ao observar as melhores correspondências de cada imagem de consulta I_q, observou-se empiricamente que as N melhores correspondências $N = \{I_1, I_2, .., I_n\}$ eram imagens quase consecutivas de I_q. Espera-se que as melhores correspondências sejam imagens consecutivas a I_q no caso de correspondência de imagens, mas não em correspondência de cenas aproximado à medida de intersecção de areas de campo de visão. Como a avaliação realizada neste trabalho utiliza a intersecção de áreas, no Gráfico 22 é possível analisar como os métodos se comportam ao rejeitarmos todas as imagens que sejam do conjunto A_s definido por:

$$A_s = \{I_{q-s}, ..., I_{q-2}, I_{q-1}, I_q, I_{q+1}I_{q+2}, ..., I_{q+s}\} \tag{4.5}$$

Utilizando o valor de $s = 10$, observou-se que todos os métodos sofrem um decréscimo em sua acurácia. O que significa que a melhor correspondência não necessariamente considera duas imagens com alto grau de intersecção de áreas, mas capturadas com alta diferença de rotação entre si como similares. Um fator que merece destaque é que o método de extração de características com métrica usando dados reais é mais robusto ao

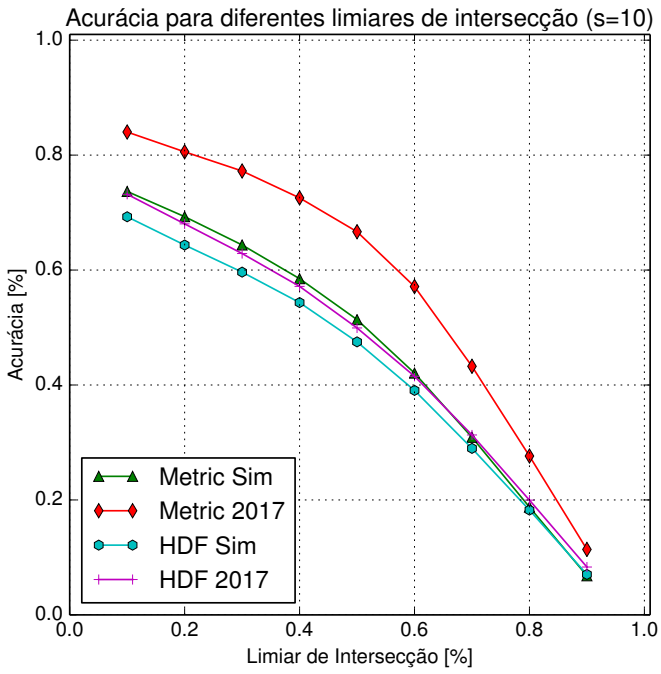


Figura 22: Acurácia para diferentes níveis de intersecções: a mesma arquitetura é treinada de quatro formas distintas. HDF Sim é treinada para regressão de Pose usando conjuntos de dados simulados e é validada com os dados de 2014. HDF 2017 é treinada para regressão de Pose usando conjuntos de dados reais do *dataset* de 2017 e validada com os dados de 2014. *Metric* Sim utiliza de aprendizagem de métrica com dados simulados e é validada com dados reais de 2014. *Metric* 2017 usa a abordagem de aprendizagem de métrica com dados de 2017 e é validada com os dados coletados em 2014. Neste caso, as primeiras 10 imagens com menor distância são rejeitadas.

recuperar imagens com no mínimo 10 capturas de diferença.

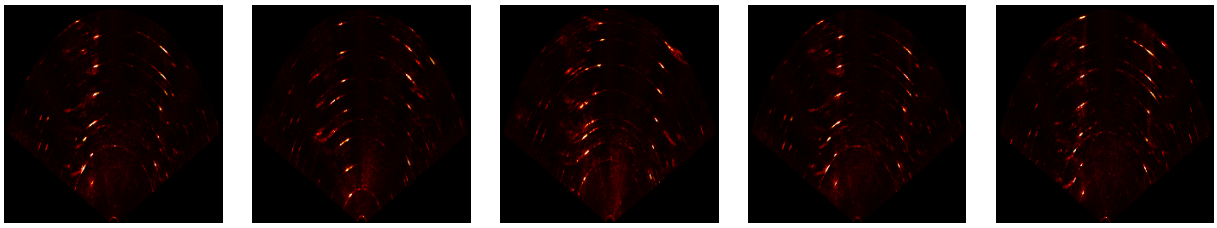
4.3.4 Avaliação Quali-Quantitativa

Nesta seção procurou-se exemplificar com casos reais como a extração de característica se comportou para diferentes os métodos. Em virtude da impossibilidade de se realizar uma análise qualitativa para cada uma das 10231 imagens capturadas, algumas imagens foram selecionadas e os dados de cada consulta foram exibidos. O único critério de rejeição utilizado foi recusar imagens I_i e I_j foi $i \neq j$, pois $L^2(F_i, F_j) = 0$ se $i = j$.

4.3.4.1 Modelos com dados reais superam modelos simulados

Na Figura 23 pode-se observar que a imagem original contém um padrão de postes em duas colunas. Esta cena captura um píer. Os modelos *HDF Sim* e *Metric Sim* encontram imagens com padrões parecidos, entretanto são imagens que não estão próximas temporalmente e não possuem grande intersecção com a imagem de consulta como pode ser visto na Tabela 4.

O modelo *HDF 2017* recuperou a imagem mais próxima temporalmente e com



(a) I_q (b) HDF Sim (c) Metric Sim (d) HDF 2017 (e) Metric 2017

Figura 23: Consulta a partir da Imagem 872

a maior intersecção, superando o modelo *Metric 2017*. Considerando que a recuperação acerta com mais de 50% de área de intersecção, ambos os modelos estariam corretos.

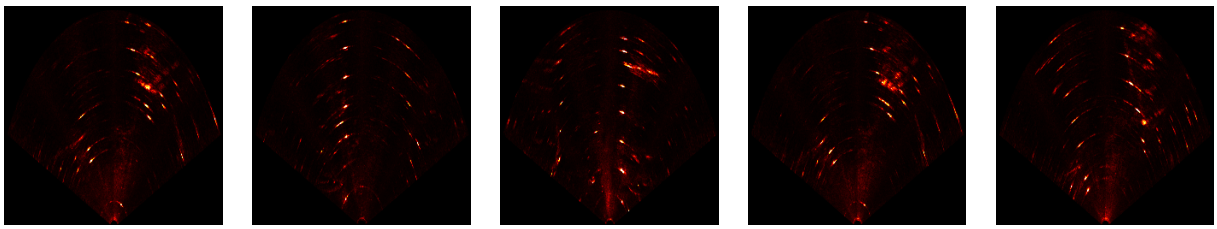
Tabela 4: Dados de Recuperação da Imagem 872

	Melhor Candidato	Área de Intersecção	L^2
HDF Sim	8253	32,21%	2,484
Metric Sim	8988	0,55%	1,616
HDF 2017	874	**93,82**%	1,311
Metric 2017	893	64,38%	3,954

É importante destacar que cada método trabalha com a distância L^2 em uma escala. Portanto, não necessariamente a menor distância L^2 retorna o menor resultado. Pois a distância L^2 de *Metric 2017* é superior à distância dos modelos simulados que fracassaram na recuperação de imagens.

4.3.4.2 Modelo HDF 2017 supera os demais modelos

Na Figura 24 pode-se observar que a imagem original contém um padrão de postes em duas colunas. Esta cena captura um píer assim como a imagem 872. Entretanto, diferentemente da imagem 872, essa imagem pode ser considerada bem difícil pois possui um número bem reduzido de características e estruturas presentes.



(a) I_q (b) HDF Sim (c) Metric Sim (d) HDF 2017 (e) Metric 2017

Figura 24: Consulta a partir da Imagem 7990

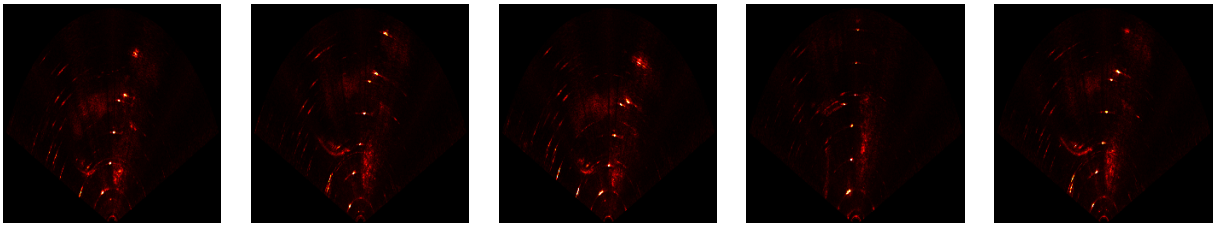
O modelo *HDF 2017* recuperou a imagem mais próxima temporalmente e com a maior intersecção, superando todos os demais modelos como pode ser observado na Tabela 5.

Tabela 5: Dados de Recuperação da Imagem 7990

	Melhor Candidato	Área de Intersecção	L^2
HDF Sim	3796	7,68%	2,543
Metric Sim	2007	46,98%	2,214
HDF 2017	7991	**98,91**%	0,953
Metric 2017	8092	75,86%	6,242

4.3.4.3 Modelos simulados contra modelos com dados reais

Este caso é menos comum conforme verificado pelos gráficos de acurácia nas Figuras 21 e Figuras 22. É esperado que modelos treinados em simulador tenham um desempenho inferior a modelos treinados com dados reais. Apesar de ser menos freqüente, na Figura 25 há uma situação contrária a esta expectativa.



(a) I_q (b) HDF Sim (c) Metric Sim (d) HDF 2017 (e) Metric 2017

Figura 25: Consulta a partir da Imagem 7646

Tabela 6: Dados de Recuperação da Imagem 7646

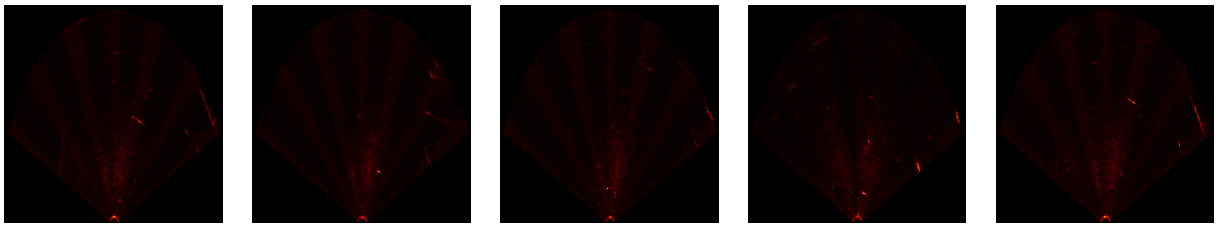
	Melhor Candidato	Área de Intersecção	L^2
HDF Sim	7616	67,08%	3,959
Metric Sim	7651	70,89%	3,836
HDF 2017	7532	30,66%	2,125
Metric 2017	7613	**79,53**%	5,896

Observa-se na Tabela 6 que todos os modelos exceto o HDF 2017 encontram uma imagem que satisfaz o critério de 50% ou mais de intersecção.

4.3.4.4 Modelo Metric 2017 supera os demais modelos

O método Metric 2017 apresentou os melhores resultados na Figura 21 e Figura 22. Na Figura 26 não são capturadas muitas estruturas, portanto os métodos precisam

comparar padrões de insonificação. Neste exemplo apenas o Metric 2017 traz uma imagem relevante.



(a) I_q (b) HDF Sim (c) Metric Sim (d) HDF 2017 (e) Metric 2017

Figura 26: Consulta a partir da Imagem 6006

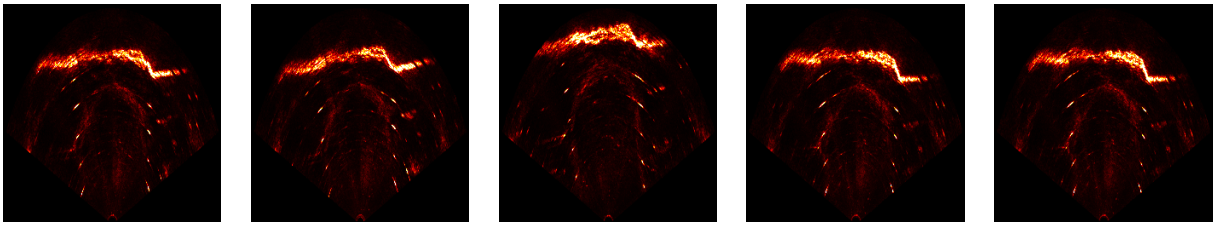
Na Tabela 7 constam os dados de busca, com o Metric 2017 trazendo uma imagem com $87,77\%$ de intersecção. Este é um exemplo que é muito arriscado efetuar uma classificação em virtude da falta de estruturas presentes nas imagens.

Tabela 7: Dados de Recuperação da Imagem 6006

	Melhor Candidato	Área de Intersecção	L^2
HDF Sim	6213	3,96%	3,196
Metric Sim	6181	7,72%	1,764
HDF 2017	9646	0%	1,472
Metric 2017	6029	**87,77%**	3,294

4.3.4.5 Imagem próxima da superfície terrestre

Quando o sonar captura uma imagem próxima à superfície terrestre, a parede de terra apresenta um padrão bastante característico. A Figura 27 demonstra um momento da missão em que o ROV estava capturando a margem da superfície terrestre.



(a) I_q (b) HDF Sim (c) Metric Sim (d) HDF 2017 (e) Metric 2017

Figura 27: Consulta a partir da Imagem 8606

Neste caso, todas as imagens são visualmente muito similares. Na Tabela 8, o modelo Metric Sim traz uma imagem muito similar a imagem de consulta. Um fator interessante é que esta imagem está localizada no início do *dataset*, portanto não faz parte das imagens consecutivas à imagem de consulta.

Tabela 8: Dados de Recuperação da Imagem 8606

	Melhor Candidato	Área de Intersecção	L^2
HDF Sim	8604	**93,04**%	1,783
Metric Sim	1232	79,38%	2,921
HDF 2017	8607	92,14%	1,153
Metric 2017	8607	92,14%	7,405

4.3.5 Considerações sobre modelos

Ao utilizar uma mesma arquitetura de CNN, mas treinada com diferentes conjuntos de dados e com diferentes estratégias, pode-se isolar questões de arquitetura às questões de modelagem do problema. Não é correto afirmar que a utilização de *triplets* (modelos Metric Sim e Metric 2017) é categoricamente superior ao uso de características da imagem referentes à posição (HDF Sim e HDF 2017). Vale lembrar que são inúmeras às variáveis a serem consideradas para modelagem de um problema. Entretanto, considerando a maneira como os *datasets* deste trabalho foram coletados e o problema que foi objeto de estudo, pode-se afirmar que o resultado obtido com o uso de *triplets* e a medida de similaridade proposta de intersecção de áreas para recuperação de imagens permite a busca baseada em conteúdo de imagens similares com maior taxa de acertos do que um trabalho anterior que utiliza diretamente a posição para treinamento da rede. A principal vantagem da

abordagem proposta é desacoplar a similaridade entre imagens da anotação geográfica direta da cena.

5 Conclusão e Trabalhos Futuros

Este trabalho discute o problema de encontrar relações entre cenas subaquáticas descritas por imagens acústicas obtidas através de um FLS com o uso de CNN. Foi feita uma revisão bibliográfica sobre CNNs aplicadas ao problema. Foi proposta uma arquitetura de rede neural para comparar imagens acústicas modelando o problema como regressão chamada *SMNet*. Foram utilizados limiares escalares para arquitetura ser aplicada no contexto de classificação binária. A arquitetura funciona como um classificador binário de pares de imagens, classificando-as como correspondentes ou não correspondentes.

Foram geradas curvas de Precisão e Revocação, para analisar o comportamento de detecção da arquitetura SMNet. A curva de Característica de Operador do Receptor foi empregada como uma análise complementar para o equílibrio do classificador para as duas classes. Também foi utilizada a métrica do Coeficiente de Correlação de Matthews (MCC) para descrever o comportamento da SMNet propostas em situações de desequilíbrio de amostras entre as classes. A SMNet também foi comparada com o GDT(MACHADO; DREWS-JR; BOTELHO, 2016) como classificador binário, levando em conta o MCC e a Precisão. Para o teste realizado, foram obtidos melhores resultados com a SMNet.

Seguindo uma modelagem diferente, foram analisadas duas abordagens de aprendizagem de características de imagens de FLS usando CNNs. Um método foi uma implementação com modificações de um trabalho de inspeção de cascos de navio com FLS(LI et al., 2016). E o outro método foi proposto neste trabalho, inspirado por trabalhos recentes com aprendizagem de métrica e *triplets*.

Com o intuito de realizar uma espécie de avaliação cruzada e avaliar potenciais sobreajustes, esses dois métodos extratores de características foram comparados em novos domínios. Os modelos foram treinados em um ambiente e avaliados em outro. Para cada método foi treinado um modelo em simulação e outro com dados reais. Foi observado que os modelos com dados reais se comportam melhor que os treinados com dados de simulação. Pode-se destacar que o método proposto recupera imagens acústicas relevantes para 85% das imagens coletadas em um ambiente desconhecido.

Em trabalhos futuros, pretende-se estudar novas estratégias de amostragem de treinamento para aprendizagem de métrica. Não houve tempo para testar todas as combinações de *Triplets* possíveis anotadas para os *datasets*. Portanto, é possível que apenas com uma anotação mais completa os resultados possam ser melhores. Também seria interessante integrar as características de imagens aprendidas em outros problemas de robótica subaquática, como um sistema de SLAM. Outro ponto interessante é a exploração da transformação rápida de Fourier para tratamento do sinal, como foi utilizado no trabalho

de Hurtós et al. (2015) para alinhamento e mosaico de imagens acústicas.

Referências

AULINAS, J. et al. Vision-based underwater slam for the sparus auv. In: *Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries. Germany*. [S.l.: s.n.], 2011. p. 171–179. Citado na página 23.

BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 39, n. 12, p. 2481–2495, 2017. Citado na página 28.

BAY, H.; TUYTELAARS, T.; GOOL, L. V. Surf: Speeded up robust features. *Computer vision–ECCV 2006*, Springer, p. 404–417, 2006. Citado 2 vezes nas páginas 24 e 39.

CAZALA, J. *Cazala Synaptic*. 2017. Https://github.com/cazala/synaptic/wiki/Architect. Acessado em: 2017-06-14. Citado na página 29.

CERQUEIRA, R. et al. Custom shader and 3d rendering for computationally efficient sonar simulation. Citado na página 52.

COATES, A.; NG, A.; LEE, H. An analysis of single-layer networks in unsupervised feature learning. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. [S.l.: s.n.], 2011. p. 215–223. Citado na página 44.

GUTH, F. A. et al. Underwater visual 3d slam using a bio-inspired system. In: IEEE. *Computing and Automation for Offshore Shipbuilding (NAVCOMP), 2013 Symposium on*. [S.l.], 2013. p. 87–92. Citado na página 23.

HAN, X. et al. Matchnet: Unifying feature and metric learning for patch-based matching. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 3279–3286. Citado na página 47.

HANLEY, J. A.; MCNEIL, B. J. The meaning and use of the area under a receiver operating characteristic (roc) curve. *Radiology*, v. 143, n. 1, p. 29–36, 1982. Citado na página 58.

HAYKIN, S. *Redes neurais: princípios e prática*. 2nd. ed. [S.l.]: Bookman Editora, 2007. Citado 3 vezes nas páginas 28, 30 e 51.

HE, K. et al. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. Citado 2 vezes nas páginas 30 e 47.

HO, K. L.; NEWMAN, P. Loop closure detection in slam by combining visual and spatial appearance. *Robotics and Autonomous Systems*, Elsevier, v. 54, n. 9, p. 740–749, 2006. Citado na página 23.

HOFFER, E.; AILON, N. Deep metric learning using triplet network. In: SPRINGER. *International Workshop on Similarity-Based Pattern Recognition*. [S.l.], 2015. p. 84–92. Citado 4 vezes nas páginas 36, 43, 44 e 49.

HOI, S. C.; LIU, W.; CHANG, S.-F. Semi-supervised distance metric learning for collaborative image retrieval and clustering. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, ACM, v. 6, n. 3, p. 18, 2010. Citado na página 42.

HOVER, F. S. et al. Advanced perception, navigation and planning for autonomous in-water ship hull inspection. *The International Journal of Robotics Research*, Sage Publications Sage UK: London, England, v. 31, n. 12, p. 1445–1464, 2012. Citado na página 23.

HUANG, T. A.; KAESS, M. Incremental data association for acoustic structure from motion. In: IEEE. *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. [S.l.], 2016. p. 1334–1341. Citado 2 vezes nas páginas 24 e 34.

HURTÓS, N. et al. Evaluation of registration methods on two-dimensional forward-looking sonar imagery. In: IEEE. *OCEANS-Bergen, 2013 MTS/IEEE*. [S.l.], 2013. p. 1–8. Citado na página 26.

HURTÓS, N. et al. Fourier-based registration for robust forward-looking sonar mosaicing in low-visibility underwater environments. *Journal of Field Robotics*, Wiley Online Library, v. 32, n. 1, p. 123–151, 2015. Citado 4 vezes nas páginas 23, 24, 33 e 76.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. Citado na página 47.

JOHANNSSON, H. et al. Imaging sonar-aided navigation for autonomous underwater harbor surveillance. In: IEEE. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. [S.l.], 2010. p. 4396–4403. Citado 2 vezes nas páginas 25 e 33.

KARPATHY, A.; FEI-FEI, L. Deep visual-semantic alignments for generating image descriptions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3128–3137. Citado na página 28.

KENDALL, A.; CIPOLLA, R. Geometric loss functions for camera pose regression with deep learning. In: *Proc. CVPR*. [S.l.: s.n.], 2017. v. 3, p. 8. Citado na página 36.

KENDALL, A.; GRIMES, M.; CIPOLLA, R. Posenet: A convolutional network for real-time 6-dof camera relocalization. In: IEEE. *Computer Vision (ICCV), 2015 IEEE International Conference on*. [S.l.], 2015. p. 2938–2946. Citado 2 vezes nas páginas 28 e 35.

KINGMA, D.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 49.

KRIZHEVSKY, A.; HINTON, G. Learning multiple layers of features from tiny images. Citeseer, 2009. Citado 2 vezes nas páginas 44 e 52.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2012. p. 1097–1105. Citado 3 vezes nas páginas 28, 36 e 50.

KULIS, B. Metric learning. *Tutorial in ICML*, 2010. Citado 2 vezes nas páginas 42 e 43.

LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, IEEE, v. 86, n. 11, p. 2278–2324, 1998. Citado 2 vezes nas páginas 29 e 44.

LI, J. et al. Utilizing high-dimensional features for real-time robotic applications: Reducing the curse of dimensionality for recursive bayesian estimation. In: IEEE. *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. [S.l.], 2016. p. 1230–1237. Citado 6 vezes nas páginas 28, 31, 37, 63, 64 e 75.

LONGARAY, L. *Desenvolvimento de datasets simulados em ambientes subaquáticos para uso em aplicações de deep learning*. 2017. https://github.com/lucaslongaray/simuladores-netuno. Trabalho de Conclusão de Curso (Engenharia de Computação). Citado 3 vezes nas páginas 13, 52 e 54.

LOWE, D. G. Object recognition from local scale-invariant features. In: IEEE. *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*. [S.l.], 1999. v. 2, p. 1150–1157. Citado 3 vezes nas páginas 24, 29 e 39.

LOWRY, S. et al. Visual place recognition: A survey. *IEEE Transactions on Robotics*, IEEE, v. 32, n. 1, p. 1–19, 2016. Citado 3 vezes nas páginas 27, 36 e 57.

MACHADO, M.; DREWS-JR, P.; BOTELHO, S. Descriçao e detecçao de regioes subaquáticas parcialmente estruturadas em imagens acústicas adquiridas por um sonar de imageamento frontal. 2016. Citado 6 vezes nas páginas 23, 24, 25, 39, 61 e 75.

MACHADO, M. et al. A topological descriptor of forward looking sonar images for navigation and mapping. In: SPRINGER. *Latin American Robotics Symposium*. [S.l.], 2016. p. 120–134. Citado na página 34.

MACHADO, M. et al. Description and matching of acoustic images using a forward looking sonar: A topological approach. *IFAC-PapersOnLine*, Elsevier, v. 50, n. 1, p. 2317–2322, 2017. Citado na página 32.

MCFEE, B.; LANCKRIET, G. R. Metric learning to rank. In: *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*. [S.l.: s.n.], 2010. p. 775–782. Citado na página 42.

NETZER, Y. et al. Reading digits in natural images with unsupervised feature learning. In: *NIPS workshop on deep learning and unsupervised feature learning*. [S.l.: s.n.], 2011. v. 2011, n. 2, p. 5. Citado na página 44.

PROTAS, E. et al. Visualization techniques applied to image-to-image translation. Brazilian Conference on Intelligent Systems - BRACIS. 2018. Citado na página 32.

RIBAS, D. et al. Slam using an imaging sonar for partially structured underwater environments. In: IEEE. *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. [S.l.], 2006. p. 5040–5045. Citado na página 23.

RIBEIRO, P. O. C. d. S. et al. Forward looking sonar scene matching using deep learning. In: IEEE. *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*. [S.l.], 2017. p. 574–579. Citado na página 32.

RIBEIRO, P. O. C. de S. et al. Underwater place recognition in unknown environments with triplet based acoustic image retrieval. 17th IEEE International Conference on Machine Learning and Applications (Submetido). 2018. Citado na página 31.

RUBLEE, E. et al. Orb: An efficient alternative to sift or surf. In: IEEE. *Computer Vision (ICCV), 2011 IEEE international conference on*. [S.l.], 2011. p. 2564–2571. Citado na página 37.

SANTOS, M. dos et al. Object classification in semi structured enviroment using forward-looking sonar. *Sensors*, v. 17, n. 10, 2017. ISSN 1424-8220. Disponível em: <http://www.mdpi.com/1424-8220/17/10/2235>. Citado na página 32.

SCHMIDHUBER, J. Deep learning in neural networks: An overview. *Neural networks*, Elsevier, v. 61, p. 85–117, 2015. Citado na página 28.

SCHROFF, F.; KALENICHENKO, D.; PHILBIN, J. Facenet: A unified embedding for face recognition and clustering. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 815–823. Citado na página 42.

SILVEIRA, L. et al. An open-source bio-inspired solution to underwater slam. *IFAC-PapersOnLine*, Elsevier, v. 48, n. 2, p. 212–217, 2015. Citado 2 vezes nas páginas 24 e 53.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado na página 30.

SONG, H. O. et al. Deep metric learning via lifted structured feature embedding. In: IEEE. *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*. [S.l.], 2016. p. 4004–4012. Citado 2 vezes nas páginas 42 e 49.

SONG, H. O. et al. Deep metric learning via lifted structured feature embedding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2016. p. 4004–4012. Citado na página 50.

SORENSON, H. W. *Kalman filtering: theory and application*. [S.l.]: IEEE, 1985. Citado 2 vezes nas páginas 23 e 64.

STURM, J. et al. A benchmark for the evaluation of rgb-d slam systems. In: IEEE. *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. [S.l.], 2012. p. 573–580. Citado na página 52.

SUNDERHAUF, N. et al. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015. Citado na página 35.

SZEGEDY, C. et al. Inception-v4, inception-resnet and the impact of residual connections on learning. In: *AAAI*. [S.l.: s.n.], 2017. v. 4, p. 12. Citado 2 vezes nas páginas 35 e 50.

SZEGEDY, C. et al. Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 1–9. Citado 2 vezes nas páginas 35 e 47.

VALDENEGRO-TORO, M. Improving sonar image patch matching via deep learning. *arXiv preprint arXiv:1709.02150*, 2017. Citado 2 vezes nas páginas 37 e 47.

VO, N. N.; HAYS, J. Localizing and orienting street views using overhead imagery. In: SPRINGER. *European Conference on Computer Vision*. [S.l.], 2016. p. 494–509. Citado 3 vezes nas páginas 36, 49 e 50.

WANG, J. et al. Learning fine-grained image similarity with deep ranking. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. [S.l.: s.n.], 2014. Citado 4 vezes nas páginas 42, 43, 44 e 45.

WEINBERGER, K. Q.; SAUL, L. K. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, v. 10, n. Feb, p. 207–244, 2009. Citado 2 vezes nas páginas 43 e 44.

WEISS, G. M.; MCCARTHY, K.; ZABAR, B. Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? *DMIN*, Citeseer, v. 7, p. 35–41, 2007. Citado na página 41.

ZAGORUYKO, S.; KOMODAKIS, N. Learning to compare image patches via convolutional neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2015. p. 4353–4361. Citado 3 vezes nas páginas 29, 36 e 37.

ZBONTAR, J.; LECUN, Y. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, v. 17, n. 1-32, p. 2, 2016. Citado na página 36.

ZHANG, C. et al. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. Citado na página 52.

ZHU, J.-Y. et al. Unpaired image-to-image translation using cycle-consistent adversarial networks. *arXiv preprint arXiv:1703.10593*, ICCV, 2017. Citado na página 28.