



Uso de Deep Learning para Codificação da Transmissão de Imagem Subaquática

Débora Debiaze de Paula

Rio Grande, 2023

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

Dissertação de Mestrado

Uso de Deep Learning para Codificação da Transmissão de Imagem Subaquática

Débora Debiaze de Paula

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande - FURG, como requisito parcial para a obtenção do grau de Mestra em Engenharia de Computação

Orientador: Prof^ª. Dr^ª. Silvia Silva da Costa Botelho
Co-orientadores: Dr. Felipe Codevilla Moraes
Prof. Dr. Ricardo Nagel Rodrigues

Rio Grande, 2023

Ficha Catalográfica

P324u Paula, Debora Debiaze de.
Uso de *Deep Learning* para codificação da transmissão de
imagem subaquática / Debora Debiaze de Paula. – 2023.
102 f.

Dissertação (mestrado) – Universidade Federal do Rio Grande –
FURG, Programa de Pós-Graduação em Computação, Rio
Grande/RS, 2023.

Orientadora: Dra. Silvia Silva da Costa Botelho.

Coorientador: Dr. Felipe Codevilla Moraes.

Coorientador: Dr. Ricardo Nagel Rodrigues.

1. Codificação de Imagens Subaquáticas 2. Autoencoder
Variacional 3. Comunicação Subaquática 4. Compressão 5. SIFT
6. *Deep Learning* I. Botelho, Silvia Silva da Costa II. Moraes, Felipe
Codevilla III. Rodrigues, Ricardo Nagel IV. Título.

CDU 004.92

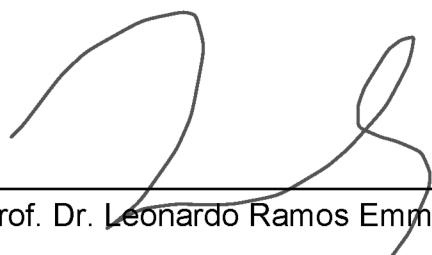
Catálogo na Fonte: Bibliotecário José Paulo dos Santos CRB 10/2344

DISSERTAÇÃO DE MESTRADO

Uso de Deep Learning para Codificação da Transmissão de Imagem Subaquática

Débora Debiaze de Paula

Banca examinadora:



Prof. Dr. Leonardo Ramos Emmendorfer (UFSM)

Documento assinado digitalmente



PAULO LILLES JORGE DREWS JUNIOR
Data: 15/02/2023 09:05:46-0300
Verifique em <https://verificador.iti.br>

Prof. Dr. Paulo Lilles Jorge Drews Junior (FURG)



Prof. Dr. Vagner Santos da Rosa (FURG)

Profa. Dra. Silvia Silva da Costa Botelho (FURG)
Orientadora

Dedico desta dissertação de mestrado ao meu Melhor Amigo e Salvador Jesus Cristo, porque sem Ele eu não teria concluído este trabalho e não teria chegado até aqui (e sei que com Ele, segundo Sua vontade, vou muito além). Obrigada pela Tua misericórdia e amor. Dedico também aos meus amados pais, José de Paula (in memoriam) e Ana Maria Pires Debiaze, pois para mim eles são exemplo de amor, persistência, resiliência, cuidado e altruísmo. Obrigada por estarem sempre do meu lado me orientando, aconselhando e incentivando. Amo vocês de todo coração.

AGRADECIMENTOS

Agradeço e novamente dedico à Jesus e minha mãe Ana este trabalho. Obrigada por estarem comigo durante toda essa caminhada do mestrado, que parte ocorreu durante a pandemia de COVID-19. O mundo inteiro foi impactado, perdas aconteceram e a ansiedade pelo desconhecido tomou conta, mas procurei encher meu coração de esperança e coragem, sempre lembrando do cuidado de Deus em tudo na minha vida: "Lancem sobre ele toda a sua ansiedade, porque ele tem cuidado de vocês"(1 Pedro 5.7). Agradeço a vocês dois por serem meu refúgio.

Agradeço também ao Cmt Solis e Ten Bruno por me receberem no Comando do 5º Distrito Naval em Rio Grande (RS), onde foi discutida a ideia inicial para este trabalho. Obrigada a minha incrível orientadora Profª. Drª. Silvia Botelho e co-orientadores, os grandes Prof. Dr. Ricardo Nagel e Dr. Felipe Codevilla por toparem entrar nesse barco comigo, não poupando esforços para me ensinar, ajudar e motivar. Um agradecimento especial a você Felipe, que mesmo sem essa incumbência disponibilizou o teu tempo para mim, sempre me ensinando e corrigindo. Aprendi muito e sou muito grata por tudo que você fez por mim. Obrigada Giulia Tondin pelo grande companheirismo durante as cadeiras do mestrado. Quero agradecer também aos meus colegas de turma e aos professores do programa de mestrado, que mesmo com aulas de forma online deram o seu melhor, dentro das condições impostas, atendendo e esclarecendo dúvidas.

Agradeço aos meus amigos e colegas por me escutarem e apoiarem nos momentos em que eu queria jogar tudo para cima. Agradeço em especial a minha amiga Bruna Guterres que me recebeu em sua casa, no período em que meus vizinhos não respeitaram, ou tiveram o mínimo de empatia no tempo em que o isolamento foi imposto a todos nós. Também sou muito grata pelo Prof. Me. Cláudio Mello e minha amiga Luciane Baldassari pela amizade e parceria no início da caminhada dos estudos para dissertação. Agradeço a Drª Paula Lopes pelas conversas, risadas, conselhos e torcida. Nos conhecemos a pouquinho tempo, mas como foi importante.

Para terminar, agradeço muito aos professores Dr. Rodrigo Guerra, Dr. Leonardo Emmendorfer, Dr. Paulo Drews e Dr. Vagner da Rosa que avaliaram meu trabalho nas fases de qualificação e defesa. Obrigada também aos meus colegas e amigos do projeto NETUNO e do PRH22-ANP.

Muito obrigada a todos que de alguma forma me ajudaram e contribuíram com a minha dissertação de mestrado.

Não fui eu que lhe ordenei? Seja forte e corajoso! Não se apavore, nem se desanime, pois o Senhor, o seu Deus, estará com você por onde você andar.
JOSUÉ 1.9 (Bíblia Sagrada NVI - Nova Versão Internacional)

SUMÁRIO

LISTA DE FIGURAS	7
LISTA DE TABELAS	10
LISTA DE ABREVIATURAS E SIGLAS	11
RESUMO	13
ABSTRACT	14
1 INTRODUÇÃO	15
1.1 Definição do Problema	20
1.2 Objetivo Geral	21
1.3 Objetivos Específicos	21
1.4 Organização da Dissertação	22
2 FUNDAMENTAÇÃO TEÓRICA	23
2.1 CODEC: Codificador/Decodificador de Imagens e Vídeos	23
2.1.1 Conceitos de Compressão da Informação	24
2.2 Comunicação Subaquática	29
2.2.1 Comunicação Acústica	30
2.3 Aprendizado de Máquina com Redes Neurais	33
2.3.1 Rede Convolucional	33
2.3.2 Rede Autoencoder	35
2.4 Extração de Conteúdo de Imagens	39
3 TRABALHOS RELACIONADOS AO USO DE DEEP LEARNING PARA CODIFICAÇÃO DE IMAGENS	40
3.1 Trabalhos com Imagens Não Subaquáticas	41
3.2 Trabalhos com Imagens Subaquáticas	44
3.3 Estruturação dos Trabalhos Relacionados	45
4 ARQUITETURA AUTOENCODER VARIACIONAL COM HYPER- PRIOR PARA CODIFICAÇÃO DE IMAGENS	47
4.1 Visão Geral	47
4.2 Arquitetura Autoencoder Variacional com Hyperprior	49

5	USO DO DESCRITOR SIFT NA QUANTIFICAÇÃO DA INFORMAÇÃO	54
5.1	Introdução ao Scale-Invariant Feature Transform (SIFT)	54
5.1.1	Localização de Patches e Keypoints	54
5.1.2	Extração de Descritores	55
5.2	Aplicação do SIFT na Quantificação da Informação	56
6	EXPERIMENTOS, TESTES E RESULTADOS	59
6.1	Plataforma CompressAI	59
6.2	Geração de Dataset de Imagem Subaquática	60
6.3	Resultados Experimentais	62
6.3.1	Avaliação da Inferência da Rede Original Autoencoder Variacional com Hyperprior em Imagens Subaquáticas (Original)	67
6.3.2	Rede Autoencoder Variacional com Hyperprior Treinada com Imagens Subaquáticas (Totalmente Treinado)	67
6.3.3	Ajuste Fino na Rede Original Pré-Treinada (Fine Tuning)	70
6.3.4	Avaliação da Variação do Lambda no Desempenho do CODEC	76
6.3.5	Ajuste Fino na Rede Original Pré-Treinada Sem e Com Inclusão do SIFT	82
7	CONSIDERAÇÕES FINAIS	87
7.1	Trabalhos Futuros	88
	REFERÊNCIAS	89

LISTA DE FIGURAS

Figura 1	Comparação entre Imagens Subaquáticas (Degradadas) e Imagens Restauradas (Sem Degradação) - (HU et al., 2018) (adaptado)	15
Figura 2	Definição do Problema (SHELAR; MAHALLE; SHINDE, 2020) (adaptada)	20
25		
Figura 4	Sistema de Comunicação Acústica Unidirecional (ANTÓNIO, 2016) (adaptado)	31
Figura 5	Sistema de Comunicação Acústica Bidirecional (ANTÓNIO, 2016) (adaptado)	31
Figura 6	Alcance Dependente da Frequência e o Sinal/Ruído (RIBEIRO, 2012)	32
Figura 7	Exemplo de uma Camada Convolutiva: A imagem de entrada de tamanho 32×32 com profundidade 3 na qual é aplicado um <i>kernel</i> de tamanho 5×5 com profundidade 3. Como resultado obtém-se uma saída de tamanho 28×28 com profundidade 1 - Retirado de (VICERI, 2020)	34
Figura 8	Esquema Básico do <i>Autoencoder</i> (AGGARWAL et al., 2018) (adaptado)	36
Figura 9	<i>Pipeline</i> de um Modelo <i>Autoencoder</i> Convolutiva - Adaptado de (PINTELAS; LIVIERIS; PINTELAS, 2021)	37
Figura 10	<i>Pipeline</i> de um Modelo <i>Autoencoder</i> Variacional - Adaptado de (ADITYA SHARMA, 2021)	38
Figura 11	Resumo de Algumas das Contribuições no Campo de Compressão de Imagens - Inspiração e Adaptação de (HU et al., 2021)	46

Figura 12	Arquitetura <i>Autoencoder</i> Variacional com <i>Hyperprior</i> . O <i>autoencoder</i> principal aprende uma representação latente da imagem, enquanto um <i>hyper-autoencoder</i> aprende uma representação <i>hyperprior</i> . A representação latente (y, z) é quantizada (Q) para criar latentes $(\hat{y})$ e hyper-latentes $(\hat{z})$, que são compactadas em um fluxo de <i>bits</i> usando um codificador aritmético (AE) e descompactadas por um decodificador aritmético (AD). O GDN e IGDN são funções avançadas de ativação. Então, o <i>autoencoder</i> principal aprende uma representação latente quantizada $\hat{y}$ da imagem x , a imagem reconstruída é denotada como $\hat{x}$ e o <i>hyper-autoencoder</i> aprende uma representação quantizada de hyper-latentes $\hat{z}$ (BALLÉ et al., 2018; QIAN et al., 2022). Os parâmetros de convolução são indicados como: número de filtros $\times$ altura de suporte do <i>kernel</i> $\times$ largura de suporte do <i>kernel</i> / <i>stride</i> para <i>down</i> ou <i>upsampling</i> , onde $\uparrow$ indica <i>upsampling</i> e $\downarrow$ <i>downsampling</i> - Imagem Retirada e Adaptada de (BALLÉ et al., 2018)	48
Figura 13	Análise do Gradiente - Retirado de (BOTAN, 2016)	55
Figura 14	Extração e Orientação de um Descritor - Retirado de (BROWN; SZELISKI; WINDER, 2005; BOTAN, 2016)	55
Figura 15	Descritor de <i>Keypoint</i> - Retirado de (TYAGI, 2019)	56
Figura 16	Valores de SIFT de Imagens Subaquáticas e Não Subaquáticas	56
Figura 17	Valores de SIFT de Acordo com cada Categoria	58
Figura 18	Amostra das Imagens Subaquáticas Presentes nos <i>Datasets</i> de Treino e Teste	61
Figura 19	Distribuição <i>Dataset</i> : Categoria <i>Só Água</i>	63
Figura 20	Distribuição <i>Dataset</i> : Treinamento	64
Figura 21	Distribuição <i>Dataset</i> : Categorias <i>Meio Termo</i> e <i>Praticamente Sem Água</i>	65
Figura 22	Distribuição <i>Dataset</i> : Categorias <i>Indoor</i> e <i>Outdoor</i>	66
Figura 23	Curva de Convergência de Treino (<i>Autoencoder</i> Variacional com <i>Hyperprior</i> Treinado com Imagens Subaquáticas)	68
Figura 24	Acompanhamento da Saída do Modelo - <i>Autoencoder</i> Variacional com <i>Hyperprior</i> Treinado com Imagens Subaquáticas	68
Figura 25	Acompanhamento da Saída do Modelo - <i>Autoencoder</i> Variacional com <i>Hyperprior</i> Treinado com Imagens Subaquáticas - Aproximado	69
Figura 26	<i>Autoencoder</i> Variacional com <i>Hyperprior</i> Treinado com Imagens Subaquáticas	70
Figura 27	Acompanhamento da Saída do Modelo - Ajuste Fino na Rede Original Pré-Treinada	71
Figura 28	Acompanhamento da Saída do Modelo - Ajuste Fino na Rede Original Pré-Treinada - Aproximado	71
Figura 29	Ajuste Fino na Rede Original Pré-Treinada	72
Figura 30	Gráfico BPPxPSNR	74
Figura 31	Imagem da Categoria <i>Só Água</i> com $\Lambda 1/20$	78
Figura 32	Imagem da Categoria <i>Meio Termo</i> com $\Lambda 1/20$	78
Figura 33	Imagem da Categoria <i>Meio Termo</i> com $\Lambda 1/5$	79
Figura 34	Imagem da Categoria <i>Praticamente Sem Água</i> com $\Lambda 1$	79
Figura 35	Gráfico BPPxPSNR - Λ s Diferentes	81

Figura 36	Gráfico BPPxPSNR - Sem e Com Inclusão do SIFT	84
Figura 37	SIFTxPSNRxBPP	85

LISTA DE TABELAS

Tabela 1	Relação entre Alcance e Largura de Banda (RIBEIRO, 2012)	32
Tabela 2	Resumo dos Símbolos	53
Tabela 3	Amostra do <i>Dataset</i> de Validação	61
Tabela 4	Parâmetros de Treinamento - <i>Autoencoder</i> Variacional com <i>Hyper-prior</i> Treinado com Imagens Subaquáticas	67
Tabela 5	Valores de PSNR e BPP dos Experimentos Apresentados	73
Tabela 6	Três Treinos com Grupos de Imagens e <i>Lambdas</i> Distintos	77
Tabela 7	Valores de PSNR e BPP dos Três Treinamentos com Diferentes <i>Lambdas</i>	80
Tabela 8	Parâmetros de Treinamento - Ajuste Fino na Rede Original Pré-Treinada Sem e Com Inclusão do SIFT	83
Tabela 9	Valores de PSNR e BPP do Treino Sem e Com Inclusão do SIFT . . .	83
Tabela 10	Análise dos <i>Frames</i> de um Vídeo	86

LISTA DE ABREVIATURAS E SIGLAS

AE	Autoencoder
ANP	Agência Nacional do Petróleo, Gás Natural e Biocombustíveis
AUV	Autonomous Underwater Vehicle
BPG	Better Portable Graphics
BPP	Bits per Pixel
CAE	Convolutional Autoencoder
CD	Compressão de Dados
CNN	Convolutional Neural Network
CR	Compression Ratio
DCT	Discrete Cosine Transform
DL	Deep Learning
DWT	Discrete Wavelet Transform
GDN	Generalized Divisive Normalization
IGDN	Inverse Generalized Divisive Normalization
IoT	Internet of Things
IoUT	Internet of Underwater Things
JPEG	Joint Photographics Experts Group
MSE	Mean Squared Error
PRH	Programa de Recursos Humanos
PSNR	Peak Signal to Noise Ratio
QS	Quantidade de SIFT
RDO	Rate Distortion Optimazation
RGB	Red Green Blue
RNA	Rede Neural Artificial
ROV	Remotely Operated Vehicle
SIFT	Scale Invariant Feature Transform

SNR	Signal/Noise Ratio
TB	Tamanho em Bytes
UUV	Underwater Unmanned Vehicle
VAE	Variational Autoencoder

RESUMO

PAULA, Débora Debiaze de. **Uso de Deep Learning para Codificação da Transmissão de Imagem Subaquática**. 2023. 101 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Diversas tarefas como a pesquisa, monitoramento dos oceanos, transporte, mineração marinha e a exploração de petróleo e gás são realizadas com o auxílio de veículos subaquáticos não tripulados, visto que sua evolução tecnológica permitiu que realizassem tarefas cada vez mais complexas e em grandes profundidades. A comunicação acústica é utilizada para estabelecer comunicação subaquática sem fio entre o veículo e a estação de apoio, onde se encontra o operador. Porém, esse tipo de comunicação é desafiadora em função da natureza do ambiente subaquático, que ocasiona limitações nos canais de comunicação devido à largura de banda limitada e atenuação de sinal, que resulta em perda de informação. Para aplicações em tempo real, essas perdas interferem no entendimento dos eventos. Uma maneira de contornar essa situação é mandar essa informação de forma compacta. Em missões submarinas, as imagens que geralmente são armazenadas possuem pouca possibilidade de uso, devido às suas características de não variação. Determinar a compressão aplicada, de acordo com a quantidade de informação presente em uma imagem é uma alternativa. Assim, nessa dissertação é proposto um estudo e aplicação da codificação de imagens subaquáticas utilizando o *Autoencoder Variacional*. A principal vantagem desta técnica de compressão de imagem é sua baixa distorção e alta taxa de *bits*. É apresentada uma alternativa para obtenção de imagens subaquáticas, devido a dificuldade de adquirir grande quantidade das mesmas. Também é discutido o uso do SIFT como uma tentativa de métrica para quantificar a variação da informação em uma imagem. Para avaliação da técnica proposta é utilizada a métrica de qualidade Relação Sinal-Ruído de Pico (PSNR), como também o *Bits por Pixel* (BPP) para avaliar o nível de compressão alcançado.

Palavras-chave: Codificação de Imagens Subaquáticas, Autoencoder Variacional, Comunicação Subaquática, Compressão, SIFT, Deep Learning.

ABSTRACT

PAULA, Débora Debiaze de. **Use of Deep Learning for Transmission Encoding of Underwater Image**. 2023. 101 f. Dissertação (Mestrado) – Programa de Pós-Graduação em Computação. Universidade Federal do Rio Grande - FURG, Rio Grande.

Several tasks such as research, ocean monitoring, transportation, marine mining, and oil and gas exploration are performed with the help of unmanned underwater vehicles, since their technological evolution have allowed them to perform complex tasks at great depths. Acoustic communication is used to establish wireless underwater communication between the vehicle and the support station where the operator is located. However, this type of communication is a challenge due to the nature of the underwater environment, that causes limitations in communication channels due to limited bandwidth and signal attenuation, which results in loss of information. For real-time applications, these losses interfere with the understanding of events. One way to get around this situation is to send this information in a compact form. In underwater missions, the images that are usually stored have little possibility of use, due to their non-variation characteristics. Determining the applied compression, according to the amount of information present in an image, is one alternative. Thus, in this dissertation a study and application of underwater image encoding using the Variational Autoencoder is proposed. The main advantage of this image compression technique is its low distortion and high bit rate. It is presented an alternative for obtaining underwater images, due to the difficulty of acquiring large amounts of them. Also is discussed the use of SIFT as an attempted metric for quantifying information variation in an image. To evaluate the proposed technique the Peak Signal-to-Noise Ratio (PSNR) quality metric is used, as well as the Bits per Pixel (BPP) to evaluate the level of compression achieved.

Keywords: Underwater Image Encoding, Variational Autoencoder, Underwater Communication, Compression, SIFT, Deep Learning.

1 INTRODUÇÃO

O oceano concentra aproximadamente 96% de toda a água presente no planeta Terra, sendo que 70% da superfície terrestre é coberta por água. O oceano é um recurso natural essencial para o desenvolvimento do mundo desde muito tempo, visto que ele é responsável por prover sustento, abrigar o maior habitat no mundo, e por meio dele rotas de transporte, comércio e estratégias militares são traçadas. Contudo, somente aproximadamente 20% do fundo do mar é conhecido, pois ele é classificado como local de difícil acesso para exploração (UITTO, 2001; OLIVEIRA, 2012).

Imagens capturadas em ambientes externos, constantemente, sofrem com a ação de partículas suspensas no ar, interagindo com os raios de luz ocasionando fenômenos como espalhamento e absorção (SONAGLIO et al., 2017; DUARTE, 2017; GAYA, 2018; CAO; PENG; COSMAN, 2018; JIANG et al., 2020). No caso de imagens obtidas em ambiente subaquáticos, percebe-se a presença de embranquecimento e baixo contraste, o que diminui seu entendimento e visibilidade. Um termo bastante utilizado para representar os efeitos causados pelas interações da luz com o meio no qual ela se propaga é *turbidez*. As imagens consideradas turvas são aquelas em que a visibilidade da cena capturada é degradada pela turbidez presente no ambiente (DUARTE et al., 2016; DUARTE, 2017). A Figura 1 exemplifica visualmente o que é a degradação de uma imagem e sua posterior restauração (HU et al., 2018), que por definição é o processo de recuperação da aparência original de uma imagem que possui algum grau conhecido (ou estimado) de degradação. Para chegar ao resultado de restauração demonstrado na Figura 1, os autores utilizaram técnicas de restauração com Redes Neurais Convolucionais (HU et al., 2018).



Figura 1: Comparação entre Imagens Subaquáticas (Degradadas) e Imagens Restauradas (Sem Degradação) - (HU et al., 2018) (adaptado)

Com o crescente uso e importância de sistemas automáticos que consigam lidar com grandes pressões, transmitir, gravar vídeos e imagens, observar, examinar e processar os dados com mais rapidez em ambientes subaquáticos, os Veículos Subaquáticos Não Tripulados, do inglês *Underwater Unmanned Vehicles* (UUVs) vem ganhando espaço de utilização (LYNN; BOHLANDER, 1999; LIMA et al., 2004; CHRIST; WERNLI, 2007; COELHO; PAPARELLI, 2010; PÉREZ et al., 2015; SONAGLIO et al., 2017). Estes veículos apresentam como vantagens o baixo custo de produção e operação, quando comparados a veículos tripulados, uma maior versatilidade e maior período de operação sem interrupções. São equipamentos que podem ser operados de maneira remota por um piloto ou são capazes de executar sua atividade de modo autônomo, caso possuam um sistema de controle embarcado. Atualmente, os veículos não tripulados estão presentes nos mais diversos segmentos, dentre os setores produtivos, a indústria petrolífera é uma grande usuária de UUVs, pois o equipamento é capaz de capturar imagens em ambientes de difícil acesso, como em águas profundas e ultra profundas, ou em tubulações e estruturas (LAMÔNICA; JÚNIOR, 2019). O veículo também pode auxiliar nas tarefas de perfuração, construção, inspeção e manutenção de estruturas submersas (PINTO, 2012). Quando se trata da realização da inspeção submarina de cascos de embarcações (HOVER et al., 2012), a utilização de mergulhadores para inspeção visual pode não assegurar que a área total do casco seja completamente examinada. Uma análise inapropriada da integridade do casco pode resultar em derramamento de óleo, mediante defeitos estruturais decorrentes de corrosão. Tal incidente gera danos ao meio ambiente, à imagem da instituição ou empresa que lida com o ambiente marinho e inclusive perda de vidas animais e humanas. Outro ponto que deve ser levado em conta é a exposição do mergulhador. Dependendo da inspeção a ser realizada, com o aumento da profundidade, a pressão exercida no corpo do mergulhador pode comprometer a sua saúde (ROCHA, 2013).

Os Veículos Subaquáticos Não Tripulados são classificados em dois grandes grupos (TAVARES, 2003; CHRIST; WERNLI, 2007; ROCHA, 2013): os Veículos Subaquáticos Operados Remotamente (*Remotely Operated Vehicles* – ROVs), e os Veículos Subaquáticos Autônomos (*Autonomous Underwater Vehicles* – AUVs). Atualmente, com o avanço e aumento de capacidade dos AUVs, a tendência é que as operações se tornem cada vez mais autônomas (HOLLINGER et al., 2012). Diferentemente do ROV, o AUV não necessita de um cordão umbilical para conexão com a superfície. Embora em alguns casos se faça necessário o uso de uma embarcação de apoio, o equipamento não sofre interação humana durante a operação de aquisição dos dados e, assim, a área de exploração é largamente expandida para operações em ambientes hostis (YUH, 2000; ALT, 2003; CHRIST; WERNLI, 2007; WYNN et al., 2014; NUNES, 2016). Com os sensores presentes no veículo, é possível realizar a aquisição de dados referenciados no espaço e no tempo, que são úteis para mapeamentos e inspeções de equipamentos e dutos submarinos, por exemplo, em profundidades nas quais a operação poderia ser

arriscada ou até inviável com veículos operados remotamente ou tripulados (NUNES, 2016). Mesmo com algumas vantagens do AUV em relação ao ROV, as operações com o veículo autônomo costumam ser críticas, seja pela segurança da operação ou complexidade da missão (HOLLINGER et al., 2012). Então, monitorar a missão na superfície torna-se necessário e importante (CHRIST; WERNLI, 2007; TEO; GOH; CHAI, 2014; ALLOTTA et al., 2016; ZHANG et al., 2019).

Neste contexto, para uma boa execução da tarefa, redução de riscos e posterior análise, o operador é dependente da boa transmissão das imagens e informações subaquáticas pelo modem acústico, seja para a superfície, ou para outros equipamentos. A imagem desempenha um papel essencial nas pesquisas de coleta e transmissão de dados subaquáticos, sendo uma importante fonte de informação qualitativa. No entanto, os robôs subaquáticos ainda encaram problemas relacionados à comunicação (ZHANG et al., 2015; SALTAREN; BARROSO; YAKRANGI, 2018), já que sua transmissão para a superfície é dificultada pela largura de banda reduzida, característica inerente a comunicação acústica subaquática. O canal de comunicação é limitado em largura de banda devido a atenuação do som pela água do mar, como também devido a interação com a superfície e o fundo do oceano. A velocidade de propagação do som ¹ na água é baixa, o que gera alta latência ² nas transmissões acústicas subaquáticas dificultando a sua utilização em tarefas críticas (III; ZORZI, 2010; YANG, 2012; MACHADO, 2016; ZANAJ et al., 2020). De modo geral, o canal de comunicação subaquático não suporta boas resoluções de imagens (ARABURA, 2012; RUBINO et al., 2015; ZHANG et al., 2019) sendo necessária a compressão dos dados da imagem antes da transmissão.

A compactação dos dados, que são transmitidos pelo modem acústico, pode reduzir o consumo energético do veículo causado pelo sistema de comunicação (HOSSEINI, 2012; SHRAVANA; VEENA, 2017; CORREA et al., 2020), como também poupar espaço de armazenamento e tempo de transmissão (MAHDI; MOHAMMED; MOHAMED, 2012; INDEX, 2016; WU; SINGHAL; KRAHENBUHL, 2018; CORREA et al., 2020). Para isso, mecanismos de Compressão de Dados (CD) vem sendo utilizados (ZHANG et al., 2019; ZHUANG et al., 2019; HE et al., 2020; KRISHNARAJ et al., 2020; YASUKAWA et al., 2021; INOUE et al., 2021). Os algoritmos responsáveis pela compressão da imagem precisam definir um codificador para compactação e um decodificador para reconstrução da imagem original (WU; SINGHAL; KRAHENBUHL, 2018). O codificador (*encoder*) e o decodificador (*decoder*) unidos constituem um *codec*, cujo principal objetivo é codificar imagens no menor número de *bits* possível (WU; SINGHAL; KRAHENBUHL,

¹A velocidade de propagação do som é definida como a distância que o som percorre (se propaga) em um determinado intervalo de tempo, sendo que ela varia de acordo com a profundidade, salinidade e temperatura.

²Latência expressa atraso. Por definição latência é a quantidade de atraso (tempo) que determinada requisição leva para ser transmitida de um ponto a outro.

2018). Quando aplicado ao trabalho com imagens subaquáticas, um dos desafios do *codec* é diminuir a taxa de *bits*³ da imagem, enquanto preserva a qualidade da imagem compactada o máximo possível. Tal ação torna o fluxo de *bits* mais adequado para transmissão subaquática de baixa taxa de *bits*. De modo geral, os algoritmos de compressão encontram um equilíbrio entre a taxa de compressão e o erro de reconstrução (WU; SINGHAL; KRAHENBUHL, 2018).

Métodos populares de compressão tais como o JPEG (*Joint Photographics Experts Group*) (WALLACE, 1991) e JPEG2000 (CHRISTOPOULOS; SKODRAS; EBRAHIMI, 2000) geralmente apresentam qualidade visual inferior, o que agrava as distorções já existentes e próprias das imagens subaquáticas. Estas técnicas foram desenvolvidas usando codificação de transformação (MISHRA; SINGH; SINGH, 2022; ILIOPOULOU, 2022), como: Transformada Discreta de Cosseno (DCT) (AHMED; NATARAJAN; RAO, 1974) e Transformada Discreta *Wavelet* (DWT) (BHAWNA, 2010), respectivamente. Posteriormente, as Redes Neurais e *Deep Learning* (DL) receberam atenção significativa (MISHRA; SINGH; SINGH, 2022) e seus avanços recentes comprovam a eficiência do aprendizado profundo na compressão de imagens (MISHRA; SINGH; SINGH, 2022), em relação aos métodos tradicionais de compressão (TODERICI et al., 2015; SENTO, 2016; THEIS et al., 2017; BALLÉ et al., 2018; ZHOU et al., 2018; MENTZER et al., 2019; CHENG et al., 2020; SUN et al., 2021). Um aspecto importante nas metodologias baseadas em DL é a escolha de sua função de perda (GHODRATI et al., 2019; BRUCKERT et al., 2021).

A Função de Perda ou *Loss Function* desempenha um papel importante em qualquer modelo DL, pois ela define um objetivo em relação ao qual o desempenho do modelo é avaliado e os parâmetros aprendidos pelo modelo são determinados, minimizando uma função de perda escolhida (GOODFELLOW; BENGIO; COURVILLE, 2016; SARAVANAN, 2021). Para exemplificar, em Mentzer et al. (MENTZER et al., 2020) o objetivo é alterado para deixar as imagens mais bonitas perceptualmente.

Dentre as técnicas de DL, particularmente o *Autoencoder* tem sido amplamente utilizado na redução de dimensionalidade para produzir representações compactas de imagens (MISHRA; SINGH; SINGH, 2022). Sun et al. e Jung (SUN et al., 2021; JUNG, 2021) destacam que os modelos de rede *autoencoder*, mais especificamente as redes *autoencoder* variacionais, vem ganhando cada vez mais espaço na área de compressão de imagens devido ao desempenho notável e bons resultados de taxa de compressão e qualidade visual da imagem alcançados (ZHOU et al., 2018; CHOI; EL-KHAMY; LEE, 2019; CHENG et al., 2019; WEN et al., 2019; BRAND; FISCHER; KAUP, 2021; SUN et al., 2021; YASUKAWA et al., 2021). O principal uso desse tipo de rede é reduzir a dimensionalidade da

³A taxa de *bits* indica a quantidade de dados que são processados em um determinado período de tempo. Quanto maior for a taxa de *bits*, melhor a qualidade da imagem. Porém, o tamanho da imagem será maior e será necessário maior espaço no disco para armazenamento, e quanto menor a taxa de *bits* menos qualidade e menor espaço de disco será utilizado.

informação que, no caso de imagens subaquáticas, resulta em transmissão dos dados de forma mais rápida sem, contudo, prejudicar a qualidade da imagem reconstruída. Neste cenário, novos métodos baseados no aprendizado eficiente de compressão de imagens subaquáticas buscam alcançar eficiência de codificação e qualidade de reconstrução da imagem, tornando-se o estado-da-arte em técnicas de codificação de imagens para compactar dados para comunicações confiáveis (YUAN et al., 2021).

Devido ao desafio que o ambiente subaquático impõe para a transmissão de informações, Ahn et al. (AHN et al., 2018) propõem que ao observar o fundo do mar, um AUV captura muitas imagens; no entanto, imagens sem criaturas ou objetos são de baixa prioridade para a missão. Para realizar uma missão de amostragem eficiente, o AUV deve selecionar apenas imagens interessantes ou importantes (AHN et al., 2018). Neste contexto, o AUV adquire imagens do fundo do mar, seleciona as imagens interessantes como aquelas que mostram a vida marinha, as comprime, reduzindo a quantidade de dados e as transmite para a embarcação de apoio da superfície. O processo de seleção evita a transmissão de imagens com pouca informação que contêm apenas areia plana no fundo do mar, por exemplo. Assim, o problema de comunicação acústica pode ser tratado a partir da seleção de imagens interessantes e consequente compressão (AHN et al., 2018).

Selecionar características importantes em uma imagem é um problema bem conhecido em computação. Na literatura, muitos métodos foram propostos para descrição de *features* (DREWS, 2008; MUKHERJEE; WU; WANG, 2015; XIAO et al., 2020; MA et al., 2021). Dentre estes, pode-se citar o SIFT (*Scale Invariant Feature Transform*), o qual se constitui em um descritor utilizado para extrair pontos característicos salientes em uma imagem, proposto por (LOWE, 2004a, 2004b), especificando e quantificando o conteúdo da mesma. Após a sua proposição várias variantes de SIFT foram apresentadas, como descrito em (WU et al., 2013a; MUKHERJEE; WU; WANG, 2015; GIVEKI; SOLTANSHAH; MONTAZER, 2017).

Neste contexto, pode-se propor uma técnica que quantifique a quantidade geral de informação em uma imagem, o que ajudará, por meio da alteração da *loss function*, o modelo de compressão a aprender se irá comprimir mais ou menos a imagem, dependendo do nível de informação quantificada.

Desta forma, com este trabalho de dissertação, busca-se estudar o problema da codificação da transmissão de imagem subaquática, com enfoque em desempenho, ou seja, se deseja reduzir tanto a complexidade no tempo, como a complexidade no espaço do algoritmo, por meio de métodos de rede de *Deep Learning*, visando principalmente métodos que trabalham com as Redes Neurais denominadas *Autoencoder (encoder-decoder)*, desenvolvendo uma *loss* que leve em consideração as características da imagem quantificada pelo SIFT.

1.1 Definição do Problema

A comunicação entre robôs e operadores ou outros robôs é particularmente desafiadora debaixo d'água, visto que em ambiente subaquático o sinal é rapidamente atenuado (FARR et al., 2010; ZHANG et al., 2019). Devido aos recursos de armazenamento limitados dos veículos subaquáticos e a largura de banda restrita da comunicação subaquática, uma alta taxa de compressão é necessária quando imagens ou vídeos são transmitidos para a superfície em tempo real. A Figura 2 descreve o contexto desta dissertação onde, a dois quilômetros de profundidade realiza-se uma missão com AUVs. Nesta profundidade localiza-se o Emissor de um modem acústico, junto com o AUV, que está transmitindo imagens com determinado tamanho de *bytes* (imagem original). O Receptor, do modem, na embarcação de controle recebe essa informação em seu tamanho original, o que auxilia a visibilidade do operador na embarcação.

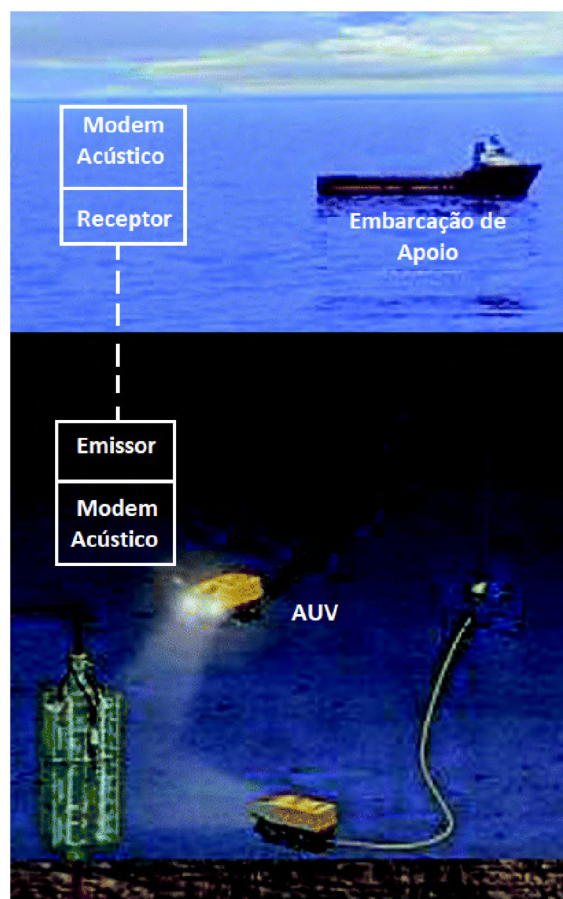


Figura 2: Definição do Problema (SHELAR; MAHALLE; SHINDE, 2020) (adaptada)

A comunicação subaquática acústica sem fio, realizada pelo modem acústico, possui algumas limitações e restrições impostas pelo canal de comunicação que são: atenuação, atraso de propagação e baixa velocidade do sinal. Enquanto que, a atenuação é a perda gradual de intensidade a medida que um sinal se move por um meio, que é maior no líquido quando comparado ao ar, o atraso de propagação está diretamente relacionado com o tempo de propagação do sinal no meio de transmissão e a baixa velocidade se refere a comunicação que é feita a uma baixa taxa de *bits*.

Os sinais acústicos se tornam a opção mais adequada para os AUVs pois, é possível estabelecer comunicação com os operadores ao longo de vários quilômetros. Já que, as frequências de comunicação embaixo d'água são de apenas dezenas de kHz ($< 50kHz$), os canais acústicos oferecem largura de banda limitada com uma taxa de dados de dezenas de *bps* (*bits* por segundo), que depende de certos fatores como o tamanho da informação a ser enviada, devido a velocidade do som na água (MURPHY, 2012; ZHANG et al., 2019). Para exemplificar, uma imagem subaquática de 1 MB (8 Mbit) transmitida em um canal acústico de baixa largura de banda, a uma taxa de ondas acústicas de 20 a 100 kbps, levará aproximadamente 80s para que seja transmitida de um veículo subaquático para uma embarcação de apoio. Portanto, comunicações acústicas de baixa largura de banda impedem a transmissão de imagens em tempo real, sendo extremamente necessário projetar algoritmos de compressão de imagem específicos que sejam adequados para transmissão em tempo real por meio de comunicação acústica subaquática, mantendo a qualidade visual da imagem (ZHANG et al., 2019). Estes algoritmos deverão considerar a quantidade de informação da imagem ao escolher o nível de compactação sem prejudicar a qualidade da imagem transmitida.

1.2 Objetivo Geral

A presente dissertação de mestrado tem como objetivo averiguar o uso de Redes *Autoencoder* no processo de codificação de imagens subaquáticas, aplicando uma função de perda que tenha em vista a quantificação de informação da imagem. O modelo deverá ser capaz de obter boas taxas de compressão associadas a boa qualidade visual da imagem, sem acentuar a degradação da imagem codificada. A proposta será validada em um conjunto de imagens reais resultantes de cenas subaquáticas.

1.3 Objetivos Específicos

Podem-se destacar os seguintes objetivos específicos:

- Estudar e avaliar os modelos das principais Redes Neurais *Deep Learning* para *codec*;
- Criar um *dataset* de imagens subaquáticas;

- Testar modelo proposto em *dataset* subaquático, analisando métricas de qualidade de imagem reconstruída e taxas de compressão;
- Analisar a utilização do SIFT como uma métrica para quantificar a diversidade de informação presente na imagem;
- Verificar a ação e comportamento do SIFT quando aplicado em imagens não subaquáticas e imagens subaquáticas;
- Incluir o SIFT na *loss* de treinamento;
- Testar modelo, com SIFT incluso, em *dataset* subaquático, analisando métricas de qualidade de imagem reconstruída e taxas de compressão.

1.4 Organização da Dissertação

Além do presente capítulo, este trabalho está dividido da seguinte maneira:

- O Capítulo 2 introduz a fundamentação teórica acerca dos principais tópicos estudados para o desenvolvimento desta dissertação;
- No Capítulo 3 é apresentada a descrição de alguns dos trabalhos relacionados ao uso de *deep learning* para codificação de imagens;
- Os Capítulos 4 e 5 apresentam a metodologia proposta, ou seja, as descrições das abordagens utilizadas no desenvolvimento deste trabalho;
- O Capítulo 6 mostra os experimentos e testes realizados, como também os resultados obtidos;
- O Capítulo 7 apresenta as considerações finais juntamente com as sugestões de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

O presente capítulo apresenta a fundamentação teórica e as ferramentas utilizadas no desenvolvimento do trabalho.

2.1 CODEC: Codificador/Decodificador de Imagens e Vídeos

Com o crescente uso das mídias digitais, tornou-se necessário contornar a dimensão dos arquivos de imagem e vídeo digitais com a adoção de padrões de compressão, que permitam a operacionalidade entre os diferentes equipamentos de diversos fabricantes (ARABURA, 2012). A compressão de dados¹ é o processo de reduzir a informação em um tamanho menor que o tamanho original (HUNG, 2007). Tal ação viabiliza a transmissão e o armazenamento dos dados de maneira mais rápida, compacta e eficiente, pois contém um número menor de dados e a "mesma" informação, quando se considera o mecanismo de compressão de dados com perdas (HUNG, 2007). Os mecanismos de compressão podem ser sem perda (*lossless*), nos quais o resultado do processo de descompressão gera dados exatamente iguais aos dados originais, ou com perda (*lossy*) onde só é possível recuperar uma aproximação dos dados originais (CORREA et al., 2020). Com relação ao desempenho, as técnicas de compressão podem ser analisadas mediante a Taxa de Compressão, do inglês *Compression Ratio* (CR) (SUBRAMANYA, 2001; RIBAS et al., 2008), dada por (1):

$$CompressionRatio = \frac{TamanhoDaImagemOriginal}{TamanhoDaImagemComprimida} \quad (1)$$

A principal meta dos algoritmos de compressão é produzir uma representação digital do sinal, para que, depois de decodificada e reproduzida, as perdas sejam minimizadas. Essas perdas devem estar dentro de determinado limiar, imperceptível ou indiferente ao receptor, buscando sempre a menor taxa de *bits* (*bit rate*) possível. Como o valor de cada *pixel* da imagem original é, normalmente, representado por um *byte*, o processo de compressão, efetivamente, ocorre quando representa-se cada *pixel* por uma quantidade menor de *bits* (ARABURA, 2012).

¹Codificar uma imagem é compactá-la em código binário (*bits*).

2.1.1 Conceitos de Compressão da Informação

Nesta subseção são apresentados alguns dos conceitos básicos que envolvem a compressão da informação.

2.1.1.1 Imagem (Frame)

Cada quadro (imagem) é uma matriz de *pixels*. Cada *pixel* representa uma tonalidade de cor e pode ter vários canais, geralmente 3 para imagens coloridas, podendo ser 1 para imagens monocromáticas (preto e branco) e 4 caso inclua a transparência, também conhecido como canal alfa. Cada quadro possui um número fixo de *pixels* na direção horizontal e vertical. O número de *pixels* em cada direção é chamado de resolução. A resolução é um valor variado, sendo que algumas resoluções são padronizadas, como por exemplo: VGA (640×480) e Full HD 1080p (1920×1080). A Resolução Espacial da imagem é determinada pelo número de *pixels* por área da imagem, ou seja, pela dimensão do *pixel* na imagem. Quanto mais *pixels* uma imagem tiver, maior é a sua resolução e melhor a sua qualidade. O processo de quantização representa o valor medido de um *pixel*, aproximando-o por meio de um valor inteiro (ARABURA, 2012; TEIXEIRA, 2014).

2.1.1.2 Vídeo

Um vídeo caracteriza-se por uma sequência de imagens (quadros) ordenadas no tempo. Para dar uma impressão de movimento a quem estiver assistindo a um vídeo, os quadros devem ser apresentados sucessivamente (TEIXEIRA, 2014). A taxa em que os quadros são apresentados é dado o nome de quadros por segundo ou *fps*, do inglês *frames per second*. A taxa mais comum usada para vídeos digitais é de 30 *fps* (TEIXEIRA, 2014). Quando a taxa de quadros ou campos é suficiente (acima de 20 *fps*), não se percebe a descontinuidade entre um quadro e outro (ARABURA, 2012).

2.1.1.3 Amostragem e Quantização

Amostragem é a escolha de um conjunto de pontos dentro de um espaço real que irão compor uma imagem digital. Definir a amostragem é conseguir selecionar pequenos conjuntos de espaços de forma que, devidamente arranjados, estes conjuntos se tornem a imagem. A amostragem está diretamente ligada com a quantidade de informação que se deseja guardar. Quanto maior a amostragem, mais detalhes se tem da imagem e consequentemente maior será o espaço necessário para o armazenamento. Para executar a amostragem sobre uma imagem real é necessário definir uma grade de amostragem. Essa grade é aplicada sobre a imagem de forma que cada célula da grade contenha uma sub-imagem (MENDES, ANDRÉ CRUZ e ALENCAR, MAYCOL DE, 2012; TEIXEIRA, 2014). Dentro de cada uma desta sub-imagens é escolhida uma cor, sendo este processo conhecido como quantização (MENDES, ANDRÉ CRUZ e ALENCAR, MAYCOL DE, 2012).

O processo de quantização considera cada uma das cores dentro de uma sub-imagem e escolhe uma. Basicamente, o que a técnica de quantização faz é buscar uma cor média que consiga representar bem as cores de um espaço delimitado (MENDES, ANDRÉ CRUZ e ALENCAR, MAYCOL DE, 2012). Ao quantizar um sinal, é inserido um erro, pois a representação do sinal passa a ser aproximada e não exata. A informação original é descartada e se guarda apenas o necessário para se fazer a operação inversa da quantização.

A quantização é um método de compressão de imagens no qual o princípio básico é dividir o fluxo de dados em blocos chamados vetores. No caso de uma imagem, o vetor é um pequeno bloco retangular ou quadrado de *pixels* de tamanho fixo chamado vetor imagem. Define-se como dicionário de codificação (*codebook*) um conjunto de vetores padrões de tamanho fixo, que está disponível na codificação e decodificação. Para cada vetor imagem de entrada, o *codebook* é consultado e o vetor padrão mais próximo é selecionado. Com isso, apenas o índice do vetor padrão é transferido. Assim, a imagem passa a ser representada por uma sequência de índices (RIBAS et al., 2008).

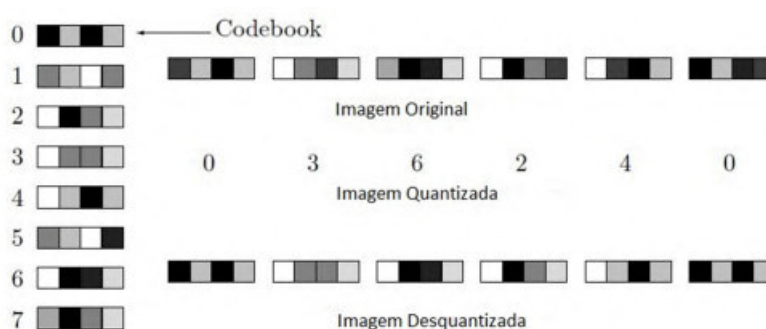


Figura 3: Processo de Quantização²

A Figura 3 mostra o exemplo de uma imagem sendo quantizada e em seguida, sendo desquantizada. Percebe-se que a imagem original de 24 *pixels*, foi segmentada em seis pedaços de 4 *pixels* consecutivos². Tendo como base de quantização (*Codebook*) os vetores representados à esquerda da imagem, foi possível quantizar a imagem gerando os códigos [0 3 6 2 4 0]. Aplicando o processo inverso (substituindo os códigos pelo vetor de *pixels*), obtém-se a imagem desquantizada. Nota-se que a imagem desquantizada é diferente da imagem original, mostrando que um erro foi introduzido no processo de quantização/desquantização².

2.1.1.4 Entropia

A Entropia de *Shannon* (SHANNON, 1948; THUM, 1984; WU et al., 2013b; PAULO, 2018) é a medida da incerteza associada a uma variável aleatória, que quantifica o valor esperado das informações contidas em uma mensagem. Neste contexto, o valor da entropia

²Parte da tese em: https://www.maxwell.vrac.puc-rio.br/37945/37945_3.PDF

pia é utilizado para medição de ordem ou quantidade da informação. Considerando uma variável discreta aleatória x que possui uma distribuição probabilística $p_x = (p_1, \dots, p_n)$, a sua quantidade de informação é dada pelo valor de entropia definido por $H(x)$ que pode ser calculado pela seguinte equação (PAULO, 2018):

$$H(x) = - \sum_{i=1}^n p_i \log_2 p_i \quad (2)$$

Quando aplicado a imagens, considera-se n como sendo o número máximo de intensidade e p_i a probabilidade de entropia para uma determinada intensidade de *pixel*, ou seja, é a probabilidade de um determinado símbolo (*pixel*) ocorrer. Assim, tem-se p_i equivalente a:

$$p_i = \frac{\text{FrequênciaDaIntensidadeDoPixel}}{\text{QuantidadeTotalDePixels}} \quad (3)$$

Portanto, baixo contraste e baixa variação de intensidade determina uma imagem com pouca informação (baixa entropia) e uma alta organização (ex: imagem toda branca), logo, sua resposta entrópica $H(x)$, tende a 0. Já imagens com alto valor de contraste e alta variação dos *pixels* (ex: imagem de uma cidade) tendem a apresentar uma resposta que pode chegar a $H(x)_{Max} = \log_2(n)$, ou seja, uma imagem com muita informação (alta entropia) (PAULO, 2018).

2.1.1.5 Compressão de Dados

A compressão de dados consiste na utilização de um conjunto de métodos com o intuito de reduzir o tamanho de um arquivo. A informação é transmitida através de símbolos de um alfabeto. No caso da escrita, o alfabeto pode ser o alfabeto latino e os símbolos são as letras. No caso de uma imagem, os *pixels* são os símbolos e o alfabeto são os valores que um *pixel* pode assumir (um caso comum é entre 0 e 255) (TEIXEIRA, 2014). Ao juntar vários símbolos se tem uma mensagem. Como exemplo, seja um arquivo texto contendo 32 caracteres:

AAAAHHHFFGGGGBBPEEEECCCCCDLLLLRR

É necessário determinar a quantidade de símbolos idênticos existentes no texto, sendo que cada um dos símbolos idênticos do texto é substituída por um número indicando a frequência do símbolo em questão (UNITINS, 2022). Assim, o texto exemplificado anteriormente seria compactado como:

5A3H1F4G2B1P3E6C1D4L2R

Essa representação compactada do exemplo possui 22 caracteres, o que resulta em uma economia de 10 *bytes* (levando em consideração que cada caractere ocupa 1 *byte* de memória).

2.1.1.6 Codificação de Huffman

Em 1952, *Huffman* propôs a primeira técnica de compressão sem perdas. Nesta técnica é calculada a frequência relativa de cada símbolo (*pixel*), que é codificado em função da sua probabilidade de ocorrência. Os símbolos que ocorrem mais frequentemente são codificados com um menor número de *bits*, enquanto os que ocorrem com menor frequência são codificados com um número de *bits* relativamente maior (GONZALEZ; WOODS, 2000; RIBAS et al., 2008). A codificação *Huffman* é considerada ótima, desde que os símbolos sejam codificados um por vez. Contudo, ela não resulta em uma alta taxa de compressão, visto que os símbolos não possuem um padrão fixo de probabilidades e isso causa uma expansão dos dados codificados.

2.1.1.7 Codificação/Decodificação Aritmética

A Codificação/Decodificação Aritmética, do inglês *Arithmetic Encoding/Decoding* é um algoritmo sem perdas (*lossless*) que usa um número baixo de *bits* para compactar dados. Como a codificação de *Huffman*, a Codificação Aritmética é uma técnica estatística. Porém, ao invés de codificar cada símbolo individualmente, uma palavra de código aritmético (em um intervalo de números reais entre 0 e 1) é atribuída a uma sequência de símbolos (SUBRAMANYA, 2001; RIBAS et al., 2008).

Uma razão pela qual os algoritmos codificação/decodificação aritmética reduzem o número de *bits* é que a mensagem é inteiramente codificada usando um único número entre 0 e 1. Cada símbolo na mensagem recebe um subintervalo no intervalo 0-1, correspondente à sua probabilidade. Para calcular a probabilidade de cada símbolo, uma tabela de frequência deve ser fornecida como entrada para o algoritmo. Esta tabela mapeia cada caractere relacionado a sua frequência. Usando a tabela de frequências, a probabilidade de ocorrência de cada símbolo é calculada. Quanto mais frequente o símbolo, menor o número de *bits* que lhe é atribuído. Como resultado, o número total de *bits* que representam a mensagem inteira é reduzido (LANGDON, 1984; WITTEN; NEAL; CLEARY, 1987; AHMED GAD, 2021).

Nesta técnica utiliza-se como entrada para a codificação apenas a mensagem que se deseja codificar e sua tabela de frequência. Já para sua decodificação é necessário o valor único que codifica a mensagem, a tabela de frequência, que deve ser idêntica a utilizada para codificação e o número de símbolos na mensagem original (AHMED GAD, 2021). Este tipo de codificação oferece maior eficiência e flexibilidade, quando comparado com outros métodos de codificação, como o *Huffman*. Porém, a codificação aritmética apresenta maior complexidade e menor possibilidade de recuperação de dados em caso de erros (SILVA, 2005).

2.1.1.8 Codecs Tradicionais

Os codecs tradicionais mais usados incluem JPEG (WALLACE, 1992), JPEG 2000 (RABBANI, 2002), WebP (GOOGLE, 2021). Desenvolvido pelo *Joint Photographic Experts Group* (JPEG), o JPEG é uma abordagem de compressão de imagem com perdas baseada em Transformada Discreta de Cosseno (DCT) codificada por *Huffman*, que foi desenvolvida há mais de três décadas. Já o JPEG 2000 foi criado com a intenção de suceder ao JPEG, usando a Transformação *Wavelet* em vez de DCT. Por fim, o WebP também tem recursos aprimorados em relação ao padrão JPEG, como codificação de previsão e quantização adaptativa, enquanto permanece computacionalmente eficiente na codificação e decodificação (AKYAZI; EBRAHIMI, 2019b).

2.1.1.9 Otimização da Taxa-Distorção

Quando se deseja decidir qual a melhor escolha para comprimir uma informação o codificador utiliza a chamada Otimização da Taxa-Distorção, do inglês: *Rate Distortion Optimazation* (RDO). De maneira simplificada, pode-se interpretar que, dado um valor de *bit-rate* limite, o codificador deve selecionar o modo de codificação que retorne a mínima distorção possível da imagem. Se S representa todos os modos de codificação possíveis e i é um elemento de S , pode-se dizer que $i \in S$.

Então:

$$\begin{aligned} i^* = \operatorname{argmin}_{i \in S} D(i) \\ R(i) \leq R_T, \end{aligned} \quad (4)$$

onde i^* é o modo ótimo que minimiza a distorção, $D(i)$ é a distorção obtida com o modo i , $R(i)$ é a taxa de *bits* do modo i e R_T é a taxa de *bits* limite (PENNY, 2016; FILHO, 2019).

2.1.1.10 Avaliação de Desempenho da Compressão

As soluções de codificação de imagem, usualmente, são avaliadas por dois aspectos: o primeiro é a eficiência de compressão, comumente medida pelo número de *bits*, quanto menos melhor. A segunda é a perda ocorrida, comumente medida pela qualidade da imagem reconstruída em comparação com a imagem original, sendo quanto maior, melhor (LIU et al., 2020).

- Bits por Pixel (BPP)

O BPP, do inglês *Bits per Pixel*, denota o número de *bits* por *pixel*, sendo a quantidade de memória considerada para armazenar um *pixel*. O número de cores diferentes em uma imagem depende do *bits* por *pixel*. Normalmente as imagens são quantizadas em 8 *bits* por *pixel*, ou seja para cada *pixel*, há uma necessidade de 8 *bits* na memória.

- Relação Sinal-Ruído de Pico (PSNR)

O PSNR, do inglês *Peak Signal to Noise Ratio*, é a métrica mais simples e comumente utilizada para avaliar a distorção de uma imagem após ser comprimida. Quanto maior o PSNR, melhor qualidade a imagem contém. Para calcular o PSNR, o MSE (Erro Quadrático Médio) é calculado em primeiro lugar. O MSE é a diferença cumulativa entre a imagem compactada e a imagem original, sendo que pequena quantidade de MSE reduz o erro e melhora a qualidade da imagem (PARMAR; PANCHOLI, 2015; JUNG, 2021).

2.2 Comunicação Subaquática

Um dos elementos mais importantes em veículos subaquáticos como os ROVs são os cabos usados para a comunicação com o mundo exterior. O sistema de comunicação entre o ROV e seus operadores é usualmente executado por cabos constituídos por fibra óptica e cabos de potência. Estes cabos, também chamados de cabos umbilicais, realizam a transmissão das informações dos sensores presentes no ROV aos operadores (FALCAO; ANTUNES, 2020). Os AUVs estendem e complementam as capacidades dos ROVs já que eles permitem uma melhor manobrabilidade, podem ser operados a uma distância maior e operam abaixo do gelo, como também em águas profundas (GIODINI; SPEK; DOL, 2015). A comunicação subaquática entre o AUV e a estação de controle é necessária para que o operador monitore o status do veículo, sendo que usualmente para esse tipo de veículo a comunicação é estabelecida através de modem acústico (CHRIST; WERNLI, 2007; MARQUES et al., 2007; RODRIGUEZ, 2016).

Os modems subaquáticos, que transmitem e recebem sinais debaixo d'água, incluem transdutores eletroacústicos para receber e transmitir sinais sonoros na água, convertendo energia elétrica em energia acústica e/ou energia acústica em energia elétrica (GIODINI; SPEK; DOL, 2015). A implementação deste tipo de tecnologia está crescendo de maneira exponencial no setor comercial e existe a necessidade de torná-la mais competitiva através da redução dos custos dos equipamentos e aumento de seu desempenho. As maiores dificuldades nesse tipo de comunicação são as características do ambiente subaquático que causam ruído, atenuam e degradam o sinal ao longo da sua propagação (MACHADO, 2016; NUNES, 2016).

A comunicação subaquática sem fio está dividida em três grupos: Comunicação Eletromagnética, Óptica e Acústica, mas este último método, na prática, é o mais viável devido aos problemas encontrados nos dois primeiros. A transmissão eletromagnética apresenta a desvantagem da alta atenuação do sinal na água, devido à absorção do meio que aumenta com a frequência. Até em baixas frequências a potência necessária para essas transmissões é inviável. Já a transmissão óptica apresenta alta taxa de transmissão com baixo consumo de energia, mas com o inconveniente do curto alcance, ocasionado pela absorção e espalhamento da luz. As aplicações, apesar de funcionarem bem em curtas distâncias, são limitadas por essa mesma característica de alcance de poucos metros, mesmo em águas límpidas, o que as torna ineficazes em operações que envolvam grandes distâncias (VASILESCU et al., 2005; RIBEIRO, 2012; MACHADO, 2016).

2.2.1 Comunicação Acústica

As comunicações acústicas subaquáticas têm sido alvo de vastas pesquisas e importantes desenvolvimentos (GIODINI; SPEK; DOL, 2015; YANG; XIAO; LI, 2021). As transmissões acústicas são diretamente influenciadas pela velocidade do som, que na água é em torno de 1.500 m/s, o que é quatro vezes mais rápido que a velocidade do som no ar. Porém, isso ainda é cinco ordens de grandeza menor do que a velocidade das ondas eletromagnéticas no ar. Apesar de as comunicações acústicas oferecerem transmissões de longo alcance (até 20 km), elas ainda possuem algumas limitações, como: a velocidade do som na água é variável e dependente da pressão (profundidade), densidade, turbulência, maré, ruído ambiente, correntes marinhas, temperatura e salinidade (URICK, 1983; LEÃO et al., 2012; RIBEIRO, 2012). A combinação destas características faz com que a direção e velocidade do som na água varie desde a superfície ao fundo, propagando-se através de caminhos curvos, devido à refração causada por camadas com diferentes velocidades (RIBEIRO, 2012; NUNES, 2016; ANTÔNIO, 2016). Mesmo com as limitações impostas pelo ambiente marinho, a implementação mais eficaz de uma comunicação subaquática é feita através de ondas acústicas (LANBO; SHENGLI; JUN-HONG, 2008; RIBEIRO, 2012).

Usualmente, a comunicação dos AUVs é estabelecida através de modem acústico (CHRIST; WERNLI, 2007; MARQUES et al., 2007; RODRIGUEZ, 2016). Este equipamento de comunicação subaquática, que é constituído por dois módulos (emissor e receptor), apresenta bom desempenho no que diz respeito a velocidade de dados e ao seu baixo consumo, permitindo estabelecer comunicação sem fio em tempo real (MACHADO, 2016). Ademais, os sistemas de comunicação acústica, de frequência por ondas de rádio e de satélite oferecem a capacidade de monitorar e redirecionar as missões de AUV em todo o mundo a partir de um navio ou de uma base *onshore* (NUNES, 2016).

A comunicação subaquática acústica entre dois pontos é dividida em dois tipos: unidirecional (Figura 4) e bidirecional (Figura 5). Na comunicação unidirecional somente um ponto é capaz de receber ondas acústicas, ao passo que na comunicação bidirecional ambos os pontos podem receber e enviar. Assim, torna-se necessário o uso de duas frequências diferentes, uma para transmissão e outra para recepção (ANTÓNIO, 2016).

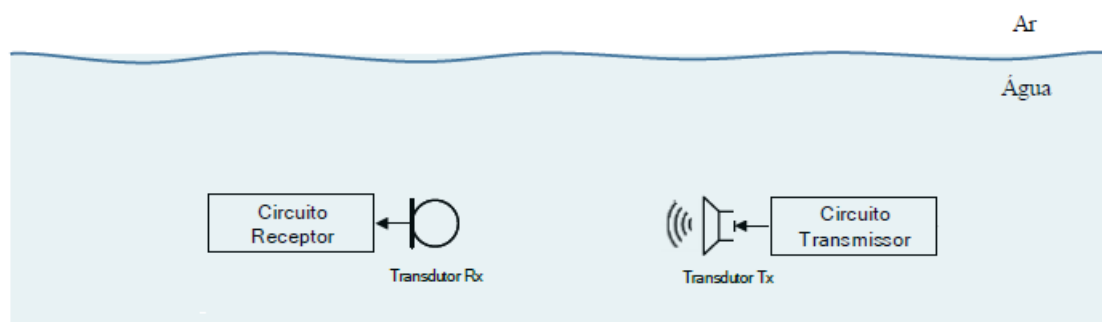


Figura 4: Sistema de Comunicação Acústica Unidirecional (ANTÓNIO, 2016) (adaptado)

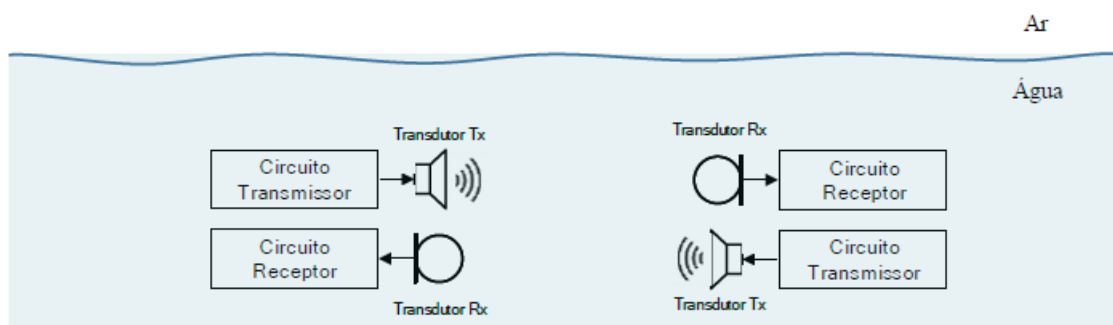


Figura 5: Sistema de Comunicação Acústica Bidirecional (ANTÓNIO, 2016) (adaptado)

2.2.1.1 Largura de Banda de um Canal Acústico

A largura de banda disponível nas comunicações subaquáticas é afetada pela variabilidade da frequência do canal acústico, que diminui com a distância. Isso acontece principalmente devido às perdas na transmissão do som e ruídos captados pelo receptor, causados principalmente pelo espalhamento da energia e absorção do som na água.

A perda de energia depende da distância entre o transmissor (emissor) e receptor, e aumenta também com a frequência (RIBEIRO, 2012). O ruído ambiente em alto mar é predominante e sua densidade espectral diminui em 20 dB/década. Assim, as dependências entre largura de banda, alcance transmissor-receptor e SNR (*Signal/Noise Ratio*) no receptor são mostradas na Figura 6 (STOJANOVIC, 2003; RIBEIRO, 2012).

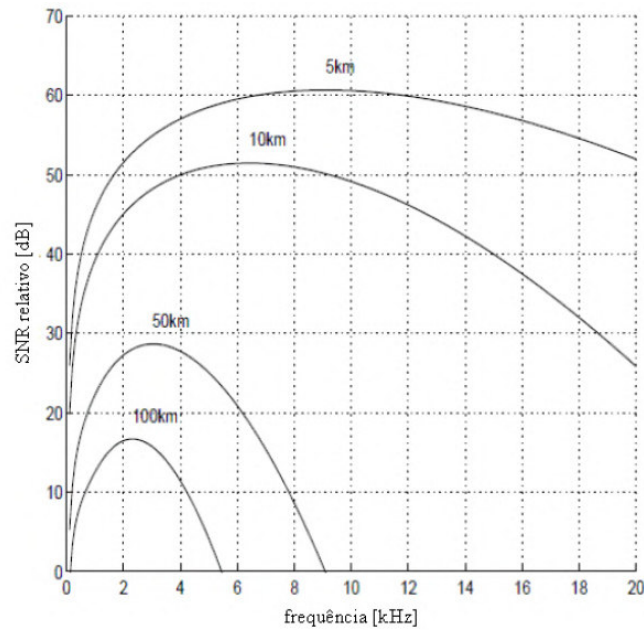


Figura 6: Alcance Dependente da Frequência e o Sinal/Ruído (RIBEIRO, 2012)

A relação entre o SNR e a frequência é apresentada na Figura 6 para quatro distâncias diferentes: 5 km, 10 km, 50 km e 100 km. Quanto maior a distância entre o transmissor e o receptor, menor é a largura de banda disponível. A Tabela 1 mostra a relação entre a largura de banda do canal acústico e a distância (STOJANOVIC, 2003; RIBEIRO, 2012).

Tabela 1: Relação entre Alcance e Largura de Banda (RIBEIRO, 2012)

Alcance (km)	Largura de Banda (kHz)
1000	Menor que 1
10 – 100	2 – 5
1 – 10	Aproximadamente 10
0,1 – 1	20 – 50
Menor que 0,1	Maior que 100

As ondas acústicas são afetadas por ruídos causados por obstáculos e turbulências. A perda pela absorção sonora é outra característica importante, fazendo com que a largura de banda e a frequência central do canal acústico sejam variáveis e diminuam com a distância. Essa limitação restringe o alcance útil para poucos quilômetros e as frequências de transmissão para menos de 30 kHz. Isto implica em taxas baixas de transmissão, normalmente em torno de 5 kbps (STOJANOVIC, 2007; RIBEIRO, 2012).

2.3 Aprendizado de Máquina com Redes Neurais

O aprendizado de máquina convencional geralmente usa métodos de otimização e descida de gradiente para aprender modelos parametrizados. Exemplos de tais modelos incluem a Redução de Dimensionalidade (HAYKIN, 2009; AGGARWAL et al., 2018). Redes neurais também são modelos parametrizados que são aprendidos com métodos de otimização contínua, os quais são representados através de grafos orientados que inspiram-se em redes de neurônios e suas conexões (AGGARWAL et al., 2018). Arquiteturas neurais complexas ou profundas tem a flexibilidade de modelar funções mais complexas com a adição de neurônios (AGGARWAL et al., 2018).

No contexto de aprendizado de máquina, surgem os conceitos de Aprendizado Supervisionado e Não Supervisionado (ISI-TICs, 2018). O aprendizado supervisionado é feito quando a partir de um conjunto de dados rotulados, previamente definidos, se encontra uma função que seja capaz de prever rótulos desconhecidos. A estimação desses rótulos pode se dar de duas formas: Classificação e Regressão. Esse tipo de aprendizagem é capaz de tomar decisões precisas quando recebe novos dados não rotulados a partir de um treinamento com dados que possuem rótulos conhecidos (ISI-TICs, 2018). Já na aprendizagem não supervisionada o conjunto de dados utilizado não possui nenhum tipo de rótulo. O objetivo desse tipo de aprendizagem é descobrir similaridades entre os objetos analisados a fim de detectar similaridades e anomalias (ISI-TICs, 2018).

Atualmente, entre as redes neurais mais amplamente utilizadas estão as chamadas Redes Neurais Convolucionais (CNN) e *Autoencoders*. O universo de possibilidades é amplo e várias outras arquiteturas especializadas em tarefas específicas vem sendo estabelecidas (Denny Ceccon, 2020). Assim, para desenvolvimento do trabalho optou-se pelo uso da rede *autoencoder* pois a mesma é capaz de aprender a reproduzir em sua saída as entradas a ela apresentadas. A rede trabalha compactando a entrada em uma representação de espaço latente, que mapeia o espaço de entrada para um espaço dimensional inferior, sem que haja grande perda de informação, e em seguida, reconstrói a saída dessa representação. A seguir, as redes Convolucionais e *Autoencoders* são apresentadas.

2.3.1 Rede Convolucional

As Redes Neurais Convolucionais, do inglês *Convolutional Neural Network* (CNN), são definidas como arquiteturas de rede similares às Redes Neurais Artificiais, porém apresentam um maior número de camadas e, conseqüentemente, de operações (GOODFELLOW; BENGIO; COURVILLE, 2016). Basicamente, as redes neurais artificiais possuem a capacidade de aprender através de exemplos e de generalizar esse aprendizado, com base nos dados, de forma a reconhecer elementos similares, imitando a capacidade humana de aprender com a experiência. Em uma Rede Convolucional, cada camada é responsável por extrair determinadas informações dos dados de entrada. A informação

flui através de cada camada da rede, com a saída da camada anterior fornecendo a entrada para a camada seguinte da rede.

A convolução tem o papel de fazer uma filtragem para extração de informações de interesse na imagem. Mais especificamente, o uso de filtros espaciais lineares é feito através de matrizes denominadas máscaras ou *kernels*. Dependendo da dimensão e dos valores presentes nessa matriz ou *kernel*, é possível obter características diferentes nas imagens como: suavização, contraste, bordas, texturas, remoção de ruídos, dentre outras. Durante a aplicação da convolução em uma imagem, o *kernel* vai se deslocando ao longo da imagem, como uma janela móvel, que vai multiplicando e somando os valores sobrepostos, segundo um número de passos determinados, conhecidos por *stride*.

Para exemplificar, a Figura 7 apresenta uma imagem de $32 \times 32 \times 3$, onde a altura e a largura têm 32 *pixels*, com 3 canais de cores, que é a profundidade da matriz. O *kernel* possui $5 \times 5 \times 3$, onde a altura e a largura têm 5 *pixels*, com a mesma profundidade da imagem, que é 3. Deslizando-se o *kernel* sobre toda a área da imagem e considerando um *stride* igual a um, tem-se uma matriz convoluída de tamanho $28 \times 28 \times 1$, pois a cada passo, é calculado o somatório do produto de todos os valores sobrepostos resultando em um único valor (VICERI, 2020).

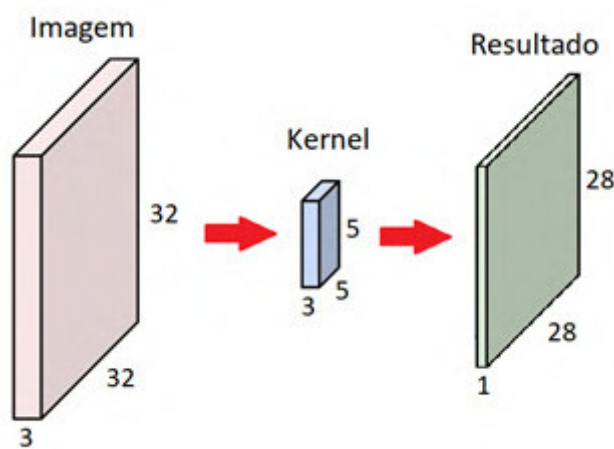


Figura 7: Exemplo de uma Camada Convolucional: A imagem de entrada de tamanho 32×32 com profundidade 3 na qual é aplicado um *kernel* de tamanho 5×5 com profundidade 3. Como resultado obtém-se uma saída de tamanho 28×28 com profundidade 1 - Retirado de (VICERI, 2020)

O que diferencia as Redes Neurais Convolucionais das Redes Neurais Artificiais clássicas é o número de camadas intermediárias ou camadas escondidas. Em resumo, cada camada da rede convolucional é responsável por uma sub-tarefa específica. As principais camadas que compõem uma CNN são (HAYKIN, 2009; AGGARWAL et al., 2018; PASSOS, BIANCA TALLITA, 2021):

- Camada de Convolução: responsável por extrair, bem como mapear o conteúdo da imagem, transformando-o em dados. Esse processo ocorre por meio da aplicação de pequenos blocos, conhecidos como filtros, que permitem a obtenção das informações de sub-blocos da imagem;
- *Pooling*: a camada de *pooling* recebe os blocos que contém as informações extraídas na etapa de convolução. Ela simplifica a informação, resumindo os dados do sub-bloco da imagem em um único valor e os repassa para uma camada totalmente conectada.

Além das camadas de convolução e *pooling*, ainda existem dois elementos fundamentais na composição de uma rede neural convolucional: camada de *dropout* e a função de ativação. A camada de *dropout* possui a responsabilidade de reduzir o *overfitting* (sobreajuste) da rede, que ocorre quando a rede decora os dados utilizados no aprendizado e não é capaz de aplicar as relações aprendidas em dados novos. É comum dizer que neste cenário o modelo gerado a partir da rede treinada não tem capacidade de generalização. Por fim, a função de ativação é responsável pela relação entre as variáveis, decidindo quais neurônios serão ativados (HAYKIN, 2009; AGGARWAL et al., 2018; PASSOS, BIANKA TALLITA, 2021).

2.3.2 Rede Autoencoder

A Rede Neural *Autoencoder* (AE) tem a capacidade de aprender a reproduzir em sua saída as entradas a ela apresentadas, ou seja, é um algoritmo de aprendizado não supervisionado, cuja a ideia é ter uma camada de saída com a mesma dimensionalidade da entrada. Se tenta reconstruir cada dimensão de forma exata, passando-a pela rede. Como o *autoencoder* realiza uma duplicação, uma réplica dos dados da entrada para a saída, ele também é chamado e conhecido por Rede Neural Replicadora. Apesar da reconstrução de dados parecer uma questão simples, pois basta apenas copiar os dados para a frente de uma camada para a outra, isso não é possível de se realizar quando a arquitetura da rede apresenta um gargalo ao fluxo da informação. Isto é, o número de unidades em cada camada intermediária é geralmente menor do que na entrada, ou saída (AGGARWAL et al., 2018).

Por consequência, essas unidades mantêm um espaço de representação reduzida dos dados e a camada final não consegue reconstruir os dados com a exatidão que deveria, causando as perdas características desse tipo de reconstrução. A função de perda (*loss*) dessa rede neural penaliza as diferenças entre a entrada e a saída, a fim de forçar a saída a ser o mais semelhante possível à entrada. A Figura 8 (b) apresenta a estrutura geral de uma rede *autoencoder*. Percebe-se que o gargalo conduz a um estrangulamento no fluxo da informação que ao ser forçado a reproduzir na saída da rede a mesma informação de entrada necessariamente extrai, em menor dimensão, as principais características dos

dados de entrada. Este espaço resultante do gargalo estrutural do fluxo é denominado de *espaço latente*. A Figura 8 (a) apresenta uma rede *autoencoder* implementada a partir de 5 camadas, três delas responsáveis pela redução dimensional (AGGARWAL et al., 2018).

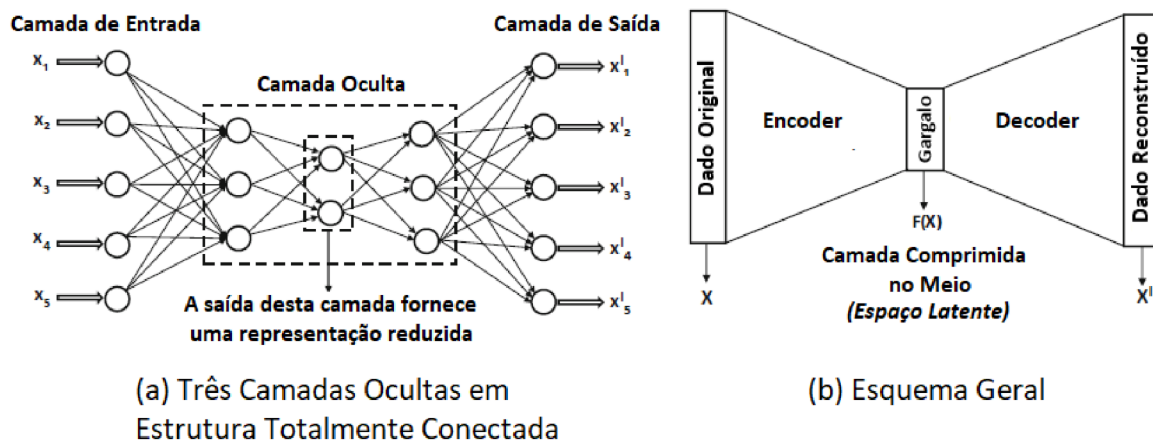


Figura 8: Esquema Básico do *Autoencoder* (AGGARWAL et al., 2018) (adaptado)

No contexto da codificação/decodificação da informação, ao observar a primeira parte da Figura 8 (b), ou seja, a parte antes do gargalo, estão os estágios da arquitetura que desempenham o papel de um codificador. Nesta perspectiva, a segunda parte após o gargalo está associada a um decodificador. O codificador (*encoder*) seria o elemento da rede responsável por extrair as correlações úteis dos atributos de entrada, comprimindo-as em uma representação de *espaço latente*. O decodificador (*decoder*) seria o elemento capaz de reconstruir/regenerar a entrada a partir da representação do espaço latente com o mínimo de perda possível. **Esses dois elementos podem ser entendidos como duas redes individuais conectadas por um ponto de conexão chamado de gargalo.** Desta forma, pode-se adotar uma rede *autoencoder* para o tratamento da compactação, através das seguintes adoções:

- *Encoder*: Conjunto de camadas da arquitetura do *autoencoder* que são responsáveis por comprimir as dimensões do espaço de entrada para as dimensões desejadas (espaço latente), ou seja é a parte da rede que compacta a entrada em uma representação de espaço latente;
- Espaço Latente: Camada que mapeia o espaço de entrada para o espaço dimensional inferior. É nele que a representação dos dados compactados é armazenada;
- *Decoder*: Conjunto de camadas do *autoencoder* que executa a recuperação de informações do espaço latente de baixa dimensão para a camada de saída, ou seja, é a parte que tem como objetivo reconstruir a entrada da representação do espaço latente.

O conceito da representação reduzida dos dados é chamado de código ou gargalo e o número de unidades nesta camada é a dimensionalidade da redução. Por existir esse gargalo não é possível mapear integralmente cada um dos atributos de entrada em sua saída relacionada. Os *autoencoders* podem ser usados tanto para compactar dados para armazenamento e/ou transmissão, quanto para representar os dados em forma reduzida para que, por exemplo, outra rede neural especializada em uma tarefa específica possa utilizá-los (Denny Ceccon, 2020). Ainda conforme (BENGIO, 2009) o treinamento básico do *autoencoder* minimiza o erro de reconstrução da imagem. Em função da aplicação desejada várias arquiteturas *autoencoders* vem sendo propostas, dentre elas pode-se citar as redes: *Autoencoder* Convolutacional e *Autoencoder* Variacional.

2.3.2.1 *Rede Autoencoder Convolutacional*

O *Autoencoder* Convolutacional (CAE) estende a estrutura básica do *autoencoder* simples (Subseção 2.3.2), alterando as camadas totalmente conectadas para camadas de convolução. A principal diferença entre o *Autoencoder* Convolutacional e o *Autoencoder* tradicional é a utilização de camadas convolucionais. Semelhante ao *autoencoder* simples, o tamanho da camada de entrada também é o mesmo das camadas de saída (ZHANG, 2018; PINTELAS; LIVIERIS; PINTELAS, 2021). Os CAEs são modelos não supervisionados de redução de dimensionalidade compostos por camadas convolucionais capazes de criar representações de imagens comprimidas, reduzir e compactar o tamanho da dimensão de entrada, remover ruídos e, ao mesmo tempo, manter todas as informações úteis da imagem (SUN et al., 2016; WICKRAMASINGHE; MARINO; MANIC, 2021; PINTELAS; LIVIERIS; PINTELAS, 2021). A Figura 9 mostra a estrutura do modelo do *Autoencoder* Convolutacional, que é composta por dois modelos CNN, o *Encoder* e o *Decoder*.

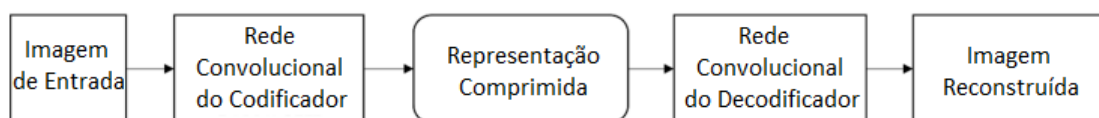


Figura 9: *Pipeline* de um Modelo *Autoencoder* Convolutacional - Adaptado de (PINTELAS; LIVIERIS; PINTELAS, 2021)

2.3.2.2 *Rede Autoencoder Variacional*

O *Autoencoder* Variacional (VAE) também é uma arquitetura, composta por um codificador e um decodificador, treinada para minimizar o erro de reconstrução entre os dados decodificados e os dados iniciais passados a rede. É um tipo de *autoencoder* que fornece uma maneira probabilística de descrever uma observação no espaço latente. Assim, em vez de construir um codificador que produza um único valor para descrever cada atri-

buto de estado latente, o codificador do VAE descreve uma distribuição de probabilidade para cada atributo latente (JORDAN, JEREMY, 2018). Assume-se que para cada ponto x amostrado da distribuição dos dados de entrada, existe também um ponto y para alguma distribuição latente (Data Science Academy, 2020; JUNG, 2021).

Os VAEs são modelados assumindo que a variável latente segue uma distribuição normal com média zero e variância unitária. Para fazer isso, as redes são treinadas para minimizar duas restrições diferentes: a primeira é a perda de distorção, que mede a distorção entre o x original e os dados $\hat{x}$ reconstruídos, que na maioria dos casos é uma perda de erro médio absoluto ou erro médio quadrático. A segunda restrição de treinamento é que a representação latente, $y = Encoder(x)$ possa ser interpretada como se y fosse amostrada de uma distribuição gaussiana normalizada (Data Science Academy, 2020; JUNG, 2021).

Na Figura 10, a imagem de entrada é fornecida ao codificador, que gera duas variáveis latentes Z_μ e Z_σ que são os parâmetros da distribuição que se aprende durante o treinamento. O vetor latente amostrado Z também pode ser chamado de camada de amostragem que amostra de um Gaussiano Multivariado $G(Z_\mu, Z_\sigma)$ onde Z_μ e Z_σ são a média e a variância, respectivamente. O vetor amostrado Z é passado para o decodificador e obtêm-se a imagem prevista $\hat{X}$ (ADITYA SHARMA, 2021). Deseja-se que o vetor latente Z siga uma distribuição normal padrão e que as variáveis Z_μ e Z_σ sejam treinadas de modo que Z_μ seja próximo de zero e Z_σ seja próximo de 1 (ADITYA SHARMA, 2021). A função de aproximação $P_\phi(Z|X)$ é o codificador probabilístico e a probabilidade condicional $P_\theta(\hat{X}|Z)$ define um modelo generativo também conhecido como decodificador probabilístico, semelhante ao decodificador do *autoencoder* simples (ADITYA SHARMA, 2021).

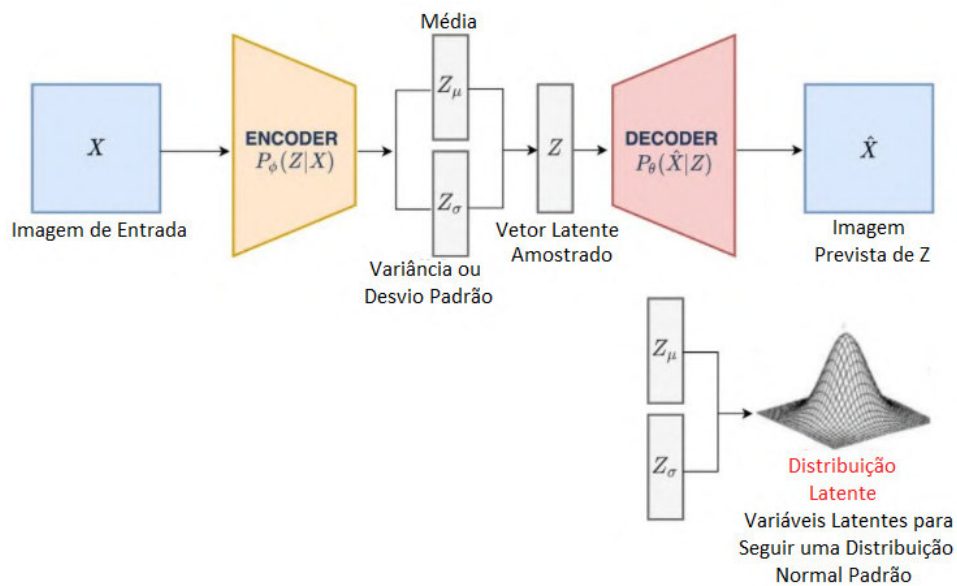


Figura 10: *Pipeline* de um Modelo *Autoencoder* Variacional - Adaptado de (ADITYA SHARMA, 2021)

2.4 Extração de Conteúdo de Imagens

O desenvolvimento de tecnologias que abordam a extração de conteúdo de imagens requer grandes quantidades de dados de treino e de tempo, para que bons modelos de extração sejam implementados (ZHANG; HSU; LEE, 2001; FAN et al., 2006; DUARTE, 2018). A extração de conteúdo em imagens e vídeos é um grande desafio para o computador, pois ele precisa ser capaz de detectar padrões e de reconhecer objetos (ZHANG; HSU; LEE, 2001; DUARTE, 2018). Definir uma função para medir a quantidade de conteúdo de uma imagem é um complicado desafio e específico da aplicação. O descritor SIFT (*Scale-Invariant Feature Transform*) tende a encontrar recursos repetíveis que são menos correlacionados à monotonia servindo como um começo aceitável para analisar e quantificar a medida de variabilidade de conteúdo presente na imagem (LOWE, 2004a; LINDBERG, 2012), detectando e descrevendo suas características locais.

O descritor SIFT localiza os pontos-chave e, em seguida, constrói descritores quantitativos associados a cada ponto-chave. O conjunto de todos os descritores e pontos-chave associados fornece um valor numérico resultante que caracteriza uma dada imagem (LOWE, 2004b; LINDBERG, 2012).

3 TRABALHOS RELACIONADOS AO USO DE DEEP LEARNING PARA CODIFICAÇÃO DE IMAGENS

Com a finalidade de melhor entender os desafios científicos da codificação da transmissão de imagem e suas atuais soluções, torna-se necessária a pesquisa e leitura acerca de trabalhos relacionados.

O uso de técnicas de aprendizado profundo, na última década, tem impactado aplicações de diversas áreas, especialmente em visão computacional e processamento de imagens, incluindo a codificação de imagem (LECUN; BENGIO; HINTON, 2015; LIU et al., 2020). Quando se trata da codificação de imagem com perdas, avalia-se tanto a eficiência de compressão, como taxa de codificação, bem como a qualidade da imagem (LIU et al., 2020). Visto que, o uso de aprendizado profundo oferece bons resultados, pesquisadores e entusiastas na área buscam verificar e analisar se o aprendizado profundo poderia favorecer a codificação de imagens. Assim, este capítulo tem seu escopo na identificação das principais abordagens atualmente propostas na área, com foco no uso de inteligência artificial, mais precisamente *deep learning* como método de compactação. A seguir, abordam-se alguns trabalhos que empregam o uso de *Deep Learning* para a codificação tanto de imagens não subaquáticas, como de imagens subaquáticas.

A partir da década de 80 até 90, o uso de RNA para codificação de imagens começa a ser utilizada (DONY; HAYKIN, 1995; JIANG, 1999; LIU et al., 2020). Porém, devido as limitações dos modelos existentes, sobretudo associados a presença de *overfitting*, ausência de grande quantidade de dados e recursos restritos de processamento, outras técnicas, na época, apresentavam melhores desempenhos (HE et al., 2016; LIU et al., 2020). Com o aumento da capacidade de processamento, o surgimento de novos modelos de redes baseados em abordagens convolucionais e a disponibilização crescente de maiores quantitativos de dados decorrente da internet, IoT (do inglês: *Internet of Things*) e transformação digital, o uso de redes profundas aparece como uma alternativa viável para o tratamento de imagens, tornando o estudo e análise da utilização do aprendizado profundo para codificação de imagem, uma área de pesquisa em desenvolvimento ativo desde 2015 (LIU et al., 2020).

3.1 Trabalhos com Imagens Não Subaquáticas

São muitos os trabalhos que apresentam abordagens baseadas em *deep learning* para a compactação fornecendo boas taxas de compressão e qualidade de reconstrução (BALLÉ; LAPARRA; SIMONCELLI, 2016c; RIPPEL; BOURDEV, 2017; THEIS et al., 2017; TODERICI et al., 2017; JOHNSTON et al., 2018; CHENG et al., 2020). Com o objetivo de produzir a redução dimensional necessária à compactação, um *autoencoder* é treinado com uma pequena camada de gargalo (RIPPEL; BOURDEV, 2017; CHENG et al., 2018; ZHOU et al., 2018; YANG et al., 2020; BRAND; FISCHER; KAUP, 2021). Alguns trabalhos codificam a imagem implementando Redes Neurais Recorrentes (RNN) (BAIG; KOLTUN; TORRESANI, 2017; TODERICI et al., 2017; JOHNSTON et al., 2018).

Toderici et al. (TODERICI et al., 2015, 2017) propõem que a compressão progressiva de imagens (ou seja, quanto mais *bits* são usados, há uma melhora na qualidade da imagem) possa ser aprendida com redes neurais recorrentes convolucionais, empregando um método de quantização durante o treinamento. Nestes trabalhos os autores buscam comprimir a imagem através da minimização da distorção para um número de *bits* específico. A imagem é codificada iterativamente e a decodificação é feita a cada etapa, para que os resíduos na iteração posterior sejam levados em conta. Assim, ocorre aumento da complexidade computacional na tarefa de compressão.

Ballé et al. (BALLÉ; LAPARRA; SIMONCELLI, 2016c, 2016b) percebe que existe uma analogia entre o problema de Otimização Taxa-Distorção e a definição de um *Autoencoder Variacional*. O *autoencoder* é treinado para minimizar a entropia dos códigos gerados pelo modelo e a distorção, ponderada por um parâmetro λ , que controla a posição do modelo na curva Otimização Taxa-Distorção (JUNG, 2021). O modelo de Ballé et al. (BALLÉ; LAPARRA; SIMONCELLI, 2016b) utiliza estruturas lineares, as chamadas GDN/IGDN (*Generalized Divisive Normalization / Inverse Generalized Divisive Normalization*) (BALLÉ; LAPARRA; SIMONCELLI, 2016a), que são camadas que favorecem a gaussianização dos dados. O GDN e IGDN são funções avançadas de ativação utilizadas nos *autoencoders*: GDN para a fase de codificação e IGDN para as camadas de decodificação, que implementam uma normalização adaptativa. Esse tipo de transformação tem se mostrado particularmente adequado para compressão de imagens (BALLÉ; LAPARRA; SIMONCELLI, 2016a, 2016b, 2016c; SHENG TAN, 2018; OLIVEIRA et al., 2021). Esta nova abordagem de normalização transforma da melhor forma possível, uma dada distribuição de probabilidade em uma distribuição normal/gaussiana, bem como fornece o par de transformada inversa que mapeia a partir da distribuição gaussiana obtida a distribuição original. Para obter a imagem normalizada, a camada GDN divide cada *pixel* por uma soma ponderada de *pixels*, na tentativa de maximizar o contraste localmente, normalizando por janelas. Já o GDN inverso (IGDN) ao invés de dividir, ele multiplica (LLOPIS; MARÍ; PÉREZ-MUELAS, 2019; JUNG, 2021). O uso desta abor-

dagem reduz o custo computacional pelo empilhamento de transformações mais simples (LAPARRA; CAMPS-VALLS; MALO, 2009; JUNG, 2021). Dando continuidade ao trabalho, (BALLÉ et al., 2018) modela melhor a distribuição latente utilizando a correlação entre os dados latentes e as imagens resultantes.

Theis et al. (THEIS et al., 2017) também utiliza como base o trabalho de Ballé et al. (BALLÉ; LAPARRA; SIMONCELLI, 2016b). Para melhorar o treinamento, os autores implementam o chamado treinamento incremental, onde o estado latente é mascarado durante o treinamento e, à medida que os limites de qualidade são atingidos, a máscara permite que cada vez mais os termos latentes sejam utilizados pela rede (JUNG, 2021). Em seu trabalho (JUNG, 2021) destaca que parte da literatura tem seguido a linha dos Modelos Variacionais, visto que os demais não apresentam resultados tão bons.

Johnson et al. (JOHNSTON et al., 2018) introduz o conceito de *priming*, onde os estados internos das camadas recorrentes são estimulados antes da codificação. Durante a codificação, eles passam a imagem original várias vezes pelo codificador, mas os códigos gerados são descartados. Isso faz com que as variáveis *ht* das camadas atinjam um estágio mais adequado para codificação. Depois de fazer isso *k* vezes, a imagem é codificada como em (TODERICI et al., 2017). O decodificador é semelhante, mas o decodificador tenta decodificar o sinal binário várias vezes, sendo sua reconstrução descartada. Uma extensão para este método, denominada de *difusão*, é fazer isso em cada iteração, superando os formatos Webp e BPG, com aumento severo do custo computacional (JUNG, 2021).

Jiang et al. (JIANG et al., 2017) propõem a compactação de imagens de alta qualidade com baixa taxa de *bits*, através do uso de duas CNNs perfeitamente integradas em uma estrutura de compactação de ponta a ponta. A primeira CNN, denominada rede neural convolucional compacta (ComCNN), aprende uma representação compacta ideal a partir de uma imagem de entrada, que preserva a informação estrutural e é então codificada usando um *codec* de imagem (por exemplo, JPEG, JPEG2000 ou BPG). A segunda CNN, denominada rede neural convolucional de reconstrução (RecCNN), é usada para reconstruir a imagem decodificada com alta qualidade no final da decodificação. Resultados experimentais demonstraram que a estrutura de compressão proposta atingiu desempenho de última geração.

Em Cheng et al. (CHENG et al., 2018) foi desenvolvida uma arquitetura de compressão de imagem com perdas, baseada em um *autoencoder* convolucional para obter codificação de alta qualidade. Inicialmente, foi projetada uma nova estrutura CAE para substituir as transformações convencionais em *codecs* tradicionais, com várias unidades de *downsampling* e *upsampling* para gerar imagens com baixas dimensões. Os resultados experimentais demonstraram que o método proposto superou o JPEG e o JPEG2000 em termos de qualidade visual.

Minnen et al. (MINNEN et al., 2018) relatam que a principal abordagem para compressão de imagens com redes neurais artificiais é aprender uma transformação não linear e um modelo de entropia fixa otimizada. O *insight* principal é que os métodos baseados em RNA existentes aprendem um modelo de entropia que é compartilhado entre o codificador e o decodificador, mas eles não transmitem nenhuma informação que permitiria ao modelo se adaptar à estrutura de uma imagem específica. O método proposto fornece desempenho de compressão de última geração em comparação com outros métodos baseados em RNA, medidos por PSNR e MS-SSIM.

Em Alexandre et al. (ALEXANDRE et al., 2019) é descrito um compressor de imagem baseado em *autoencoder* com conceito de mapas de importância para alocação de *bits* e controle de taxa. O *autoencoder* descrito usa os blocos residuais com *skip connections* para reduzir a correlação entre os *pixels* da imagem e condensar a imagem de entrada em um conjunto de mapas de recursos, uma representação compacta da imagem original. A alocação de *bits* e o controle da taxa de *bits* são implementados usando os mapas de importância e o quantizador. Embora, o método supere significativamente o JPEG, os autores destacam que ainda há uma lacuna de desempenho em relação ao BPG, sugerindo espaço para melhorias adicionais.

Em Huszar et al. (HUSZAR et al., 2020) é apresentada uma nova abordagem onde admitem-se mudanças mínimas para treinar *autoencoders* profundos. A rede proposta é computacionalmente eficiente por possuir uma arquitetura de *subpixel* que a torna apropriada para uso em imagens de alta resolução. Por fim, em (EUSÉBIO; ASCENSO; PEREIRA, 2021) é proposto melhorar a eficiência de compressão de imagem projetando e otimizando uma estrutura de codificação de imagem em que um *codec* de imagem padrão, por exemplo, JPEG é combinado com pré e pós-processamento baseado em rede neural profunda. Enquanto o pré-processamento da CNN visa simplificar a imagem para torná-la mais acessível à compressão, a CNN pós-processamento visa melhorar a imagem decodificada. Para otimizar o desempenho da compressão, as CNNs de processamento são treinadas envolvendo um terceiro CNN, denominado CNN-FakeCodec, que visa modelar a saída do *codec* de imagem. Os resultados de desempenho mostram que as soluções de codificação propostas podem superar tanto os métodos clássicos quanto os baseados em CNN.

3.2 Trabalhos com Imagens Subaquáticas

Em Krishnaraj et al. (KRISHNARAJ et al., 2020) é proposto um modelo de aprendizado profundo baseado em transformada *wavelet* discreta (DWT) para compressão de imagem em IoUT (*Internet of Underwater Things*), onde C-CNN (para compactação) e R-CNN (para reconstrução) tentam reconstruir a imagem como idêntica à imagem de entrada. O DWT pode ser aplicado como um *codec* de imagem depois do processo convolucional para minimizar o tamanho da imagem de entrada (CHANG; GIROD, 2007). O modelo compreende duas CNNs, uma na estação submarina e outra na de superfície. Após a C-CNN, um *codec* de imagem baseado em DWT também é usado. No nível do transmissor, ou seja, os nós sensores ou dispositivos IoUT no subaquático, a imagem passa por uma representação compacta, codificada utilizando a C-CNN para manter as informações estruturais da imagem e ajuda a recuperar com precisão as imagens em alta qualidade. No nível do receptor, ou seja, estação de superfície, a R-CNN é novamente aplicada para decodificar e aumentar a qualidade da imagem reconstruída. As duas CNNs são incorporadas para obter uma melhor qualidade da imagem após a compressão.

Yasukawa et al. (YASUKAWA et al., 2021) propõem um *autoencoder* convolucional como método de compressão para o processo de transmissão de imagens subaquáticas. Inicialmente, os dados de imagem adquiridos são compactados em códigos (assumindo o codificador no AUV). Na representação compactada é gerada uma perda de dados, descartando uma quantidade limitada de dados dos códigos, ou seja, algumas linhas ou colunas são eliminadas. Esses dados são enviados para o decodificador que extrai informações relevantes e reconstrói a imagem semelhante à original. O método foi avaliado usando métricas de qualidade como PSNR, concluindo que o modelo foi capaz de atingir uma relação sinal/ruído bastante alta com perda de dados proporcionando uma boa qualidade de reconstrução de imagem.

3.3 Estruturação dos Trabalhos Relacionados

A tabela na Figura 11 apresenta um resumo de algumas das principais contribuições na área da compressão de imagens no decorrer dos anos, onde é possível ver, para cada trabalho destacado, o fundamento do método implementado, medidas de desempenho das redes e os resultados obtidos.

Após a pesquisa na literatura por trabalhos acerca do uso de *Deep Learning* para compressão de imagens, foi possível perceber que grande parte dos trabalhos fundamentam seus estudos em imagens não subaquáticas. Destaca-se que, a grande maioria dos trabalhos relacionados tem foco no uso de diferentes modelos e ferramentas do aprendizado profundo para que, mesmo após o processo de codificação da informação, seja mantida a qualidade de reconstrução da imagem com uma boa taxa de compressão. Contudo, pode-se destacar o grande uso das redes neurais recorrentes: (TODERICI et al., 2015, 2017; JOHNSTON et al., 2018), como também das redes neurais *autoencoders*: (RIPPEL; BOURDEV, 2017; THEIS et al., 2017; JIANG et al., 2017; BALLÉ et al., 2018; MENTZER et al., 2018; ALEXANDRE et al., 2019; YANG et al., 2020; BRAND; FISCHER; KAUP, 2021) para o propósito de compressão de informação.

Com relação as arquiteturas de rede empregadas nos trabalhos com imagens subaquáticas, destaca-se o uso da rede *autoencoder*, como descrito em ambos os trabalhos (KRISHNARAJ et al., 2020; YASUKAWA et al., 2021), onde tem-se a ideia de que, primeiramente a imagem é codificada e passa a ser retratada por meio de uma representação compacta, onde informações estruturais da mesma são mantidas para posterior reconstrução. No final do processo, é no decodificador que a imagem é restaurada, possuindo qualidade visual e alta taxa de compressão. Vale a pena ressaltar que, os trabalhos aqui citados comprimem a imagem como um todo, não atentando para regiões ou características da imagem que possam ser mais comprimidas.

Assim, o que difere, o presente trabalho, dos trabalhos aqui relacionados, é que existem situações onde vale a pena comprimir mais determinada imagem do que outra, como por exemplo imagens que contenham fundo d'água. Imagens mais monótonas fornecem menos informação relevante ao operador para o entendimento da cena. Assim, busca-se estudar e analisar uma forma de, além de manter um baixo BPP e boa qualidade de reconstrução, comprimir mais as imagens que não tenham grande variedade e diversidade de informação e conteúdo.

Método Proposto e Referência	Título do Artigo	Medidas de Desempenho	Resultados	Ano
IMAGENS NÃO SUBAQUÁTICAS				
Variable-Rate RNN (TODERICI et al., 2015)	Variable Rate Image Compression with Recurrent Neural Networks	bpp = 0,5	O modelo exibe níveis de qualidade perceptiva que são iguais ou melhores que JPEG e WebP com uma taxa de bits média de 4% a 12% menor	2015
GDN Network (BALLÉ; LAPARRA; SIMONCELLI, 2016c)	End-to-End Optimized Image Compression	bpp = 0,11	O método proposto otimizado para uma taxa de bits baixa, em comparação com JPEG/JPEG 2000 compactadas em taxas de bits iguais ou superiores, resultou em imagem qualitativamente melhor	2016
Real-Time Adversarial (RIPPEL; BOURDEV, 2017)	Real-Time Adaptive Image Compression	bpp = 0,25	Os exemplos apresentados representam a compactação em cerca de 250 vezes. Para cada codec foi apresentada a reconstrução para uma menor taxa de bits	2017
Compressive Autoencoder (THEIS et al., 2017)	Lossy Image Compression with Compressive Autoencoder	bpp = 0,4 PSNR = 29 dB	Conforme os resultados quantitativos, em baixas taxas de bits, as reconstruções com JPEG 2000 parecem visualmente mais semelhantes às reconstruções CAE. Contudo, os artefatos produzidos pelo CAE são mais suaves e não tão ruidosos	2017
CNN (JIANG et al., 2017)	An End-to-End Compression Framework Based on Convolutional Neural Networks	bpp = 0,204 PSNR = 31,14	A estrutura de compressão proposta foi testada em diferentes taxas de bits (de 0,1bpp a 0,4bpp) e comparada com JPEG 2000. A estrutura proposta alcança um desempenho melhor ao relação ao JPEG2000, especialmente em taxas de bits muito baixas	2017
Full-Resolution RNN (TODERICI et al., 2017)	Full Resolution Image Compression with Recurrent Neural Networks	bpp = 0,5 PSNR = 33,59 dB	Houve melhoria de 4,3%-8,8% AUC (área sob a curva taxa-distorção), dependendo da métrica perceptual usada	2017
Priming RNN (JOHNSTON et al., 2018)	Improved Lossy Image Compression with Priming and Spatially Adaptive Bit Rates for Recurrent Networks	bpp = 0,2340 PSNR = 32,99 dB	Melhora da curva taxa-distorção, superando métodos baseados em rede neural e codecs de imagem padrão como o WebP, JPEG2000, e JPEG. A taxa de compressão alcançada foi menor com o método proposto	2018
3D-CNN Entropy Model (MENTZER et al., 2018)	Conditional Probability Models for Deep Image Compression	bpp = 0,15	O método supera o BPG, JPEG e JPEG2000 e nota-se que certas estruturas da imagem são bem preservadas. Além disso, o modelo de contexto reduz a taxa de bits das imagens em cerca de 10%	2018
Context Model (MINNEN; BALLÉ; TODERICI, 2018)	Joint Autoregressive and Hierarchical Priors for Learned Image Compression	bpp = 0,2 PSNR = 28 dB	O método oferece alta qualidade visual, quando comparado com outros codecs. O BPG, por exemplo, introduz alguns artefatos geométricos em algumas partes da imagem, enquanto preserva outras (9,4% mais bits que a codificação proposta)	2018
Autoencoder (ALEXANDRE et al., 2019)	An Autoencoder-Based Learned Image Compressor: Description of Challenge Proposal by NCTU	bpp = 0,126 PSNR = 26,30 dB	Autoencoder com algumas skip connections treinadas em conjunto para uma soma ponderada de MSE e MS-SSIM com um parâmetro de taxa	2019
Context Model (LEE; CHO; BEACK, 2019)	Context Adaptive Entropy Model for End-to-End Optimized Image Compression	bpp = 0,3 PSNR = 31 dB	Os ganhos de compressão, do método proposto, em termos de BD-rate (que permite medir a redução da taxa de bits oferecida por um codec) sobre JPEG2000 e BPG são 34,08% e 6,85%, respectivamente	2019
CNN (CHENG et al., 2020)	Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules	bpp = 0,519 PSNR = 33,62 dB	O método supera em desempenho os padrões de codificação como HEVC, JPEG2000 e JPEG	2020
Variable-Rate Autoencoder (YANG et al., 2020)	Variable Rate Deep Image Compression with Modulated Autoencoder	bpp = 0,1 PSNR = 25 dB	O MAE supera o desempenho dos métodos de codificação JPEG, JPEG2000 e BPG especialmente para baixas taxas de bits	2020
Autoencoder with Conditional Hyperprior (BRAND; FISCHER; KAUP, 2021)	Rate-Distortion Optimized Learning-Based Image Compression Using an Adaptive Hierarchical Autoencoder With Conditional Hyperprior		Com o modelo proposto é possível manter a qualidade da imagem e em média é possível economizar 7,4% de taxa de bits em todo o conjunto de dados	2021
IMAGENS SUBAQUÁTICAS				
DWT CNN (KRISHNARAJ et al., 2020)	Deep learning Model for Real-Time Image Compression in Internet of Underwater Things (IoUT)	bpp = 0,34	O modelo mantém eficientemente a relação entre a taxa de compressão e a qualidade da imagem reconstruída, alcançando uma economia de espaço média de 79,70% e uma relação sinal-ruído de pico médio (PSNR) de 53,961	2020
Convolutional Autoencoder (YASUKAWA et al., 2021)	Underwater Image Reconstruction Using Convolutional Auto-Encoder	PSNR = 30 dB	O modelo proposto alcança uma alta relação sinal/ruído que resulta em melhor qualidade de imagem. Os resultados indicam que o modelo CAE é capaz de proporcionar uma melhor qualidade de reconstrução e atingir uma precisão de 55 a 70%	2021

Figura 11: Resumo de Algumas das Contribuições no Campo de Compressão de Imagens - Inspiração e Adaptação de (HU et al., 2021)

4 ARQUITETURA AUTOENCODER VARIACIONAL COM HYPERPRIOR PARA CODIFICAÇÃO DE IMAGENS

Com base no levantamento bibliográfico realizado, identifica-se nas Redes *Autoencoders* uma estratégia metodológica possível de permitir a codificação/decodificação da informação no caso de compactação de imagens subaquáticas. Neste contexto, propõe-se averiguar o funcionamento da Rede *Autoencoder* Variacional como método de redução dimensional, visando seu uso em compressão de dados. A seguir apresenta-se a rede a qual servirá como base para a presente dissertação.

4.1 Visão Geral

A Figura 12 apresenta o *Autoencoder* Variacional com *Hyperprior* desenvolvido por (BALLÉ et al., 2018).

Resumidamente, a transformação de análise g_a é um codificador que transforma o vetor de imagem x em uma representação latente y com uma distribuição de probabilidades composta pela predição de desvio padrão e média zero por imagem. A saída do codificador g_a é repassada a transformada de análise *hyperprior* h_a , resumindo a distribuição dos desvios padrão em z . O z é então quantizado, comprimido e transmitido como informação secundária, que auxilia na compressão e descompressão da imagem. O codificador então usa o vetor quantizado $\hat{z}$, onde os valores são discretos, para estimar $\hat{\sigma}$, a distribuição espacial do desvio padrão, e o usa para comprimir e transmitir a representação da imagem quantizada $\hat{y}$. O decodificador primeiro recupera $\hat{z}$ do sinal comprimido. Em seguida, usa a transformada de síntese *hyperprior* h_s para obter $\hat{\sigma}$, que fornece as estimativas de probabilidade corretas para recuperar $\hat{y}$ com sucesso. Em seguida, alimenta $\hat{y}$ na transformação de síntese g_s que é um decodificador que recupera $\hat{y}$ do sinal comprimido para então por fim, recuperar a imagem reconstruída $\hat{x}$.

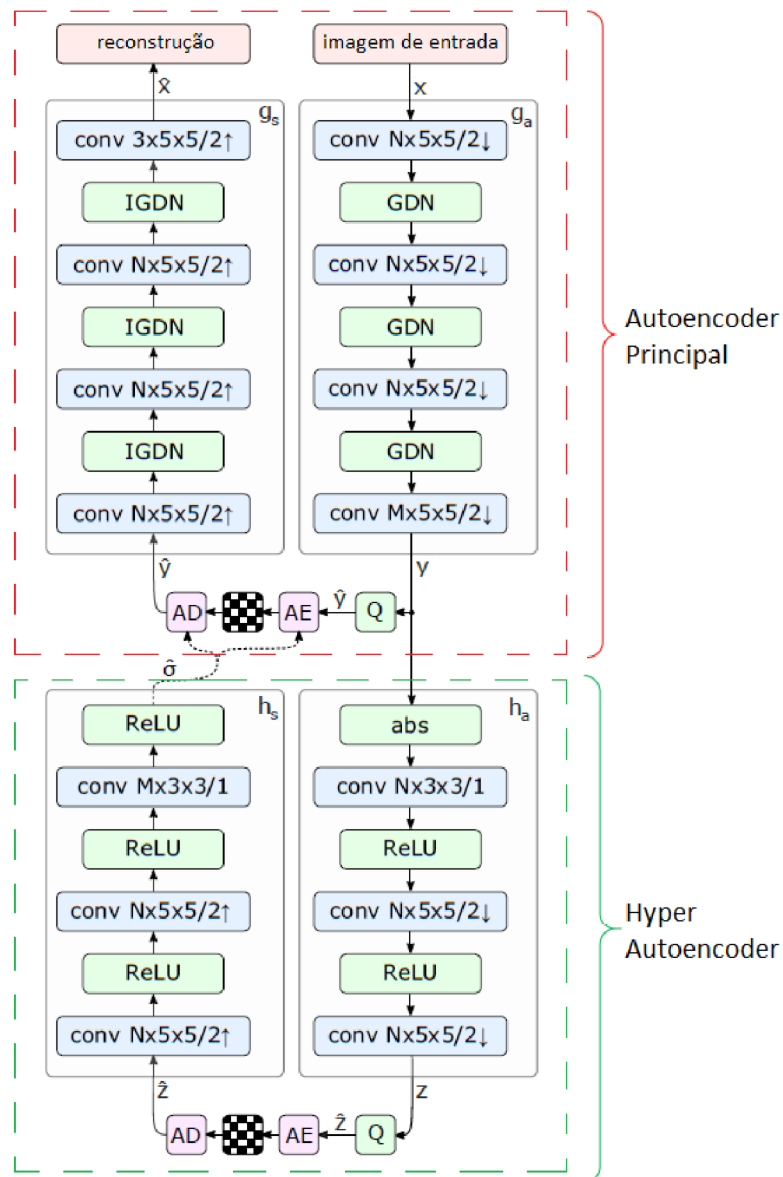


Figura 12: Arquitetura *Autoencoder* Variacional com *Hyperprior*. O *autoencoder* principal aprende uma representação latente da imagem, enquanto um *hyper-autoencoder* aprende uma representação *hyperprior*. A representação latente (y, z) é quantizada (Q) para criar latentes $(\hat{y})$ e hyper-latentes $(\hat{z})$, que são compactadas em um fluxo de *bits* usando um codificador aritmético (AE) e descompactadas por um decodificador aritmético (AD). O GDN e IGDN são funções avançadas de ativação. Então, o *autoencoder* principal aprende uma representação latente quantizada $\hat{y}$ da imagem x , a imagem reconstruída é denotada como $\hat{x}$ e o *hyper-autoencoder* aprende uma representação quantizada de hyper-latentes $\hat{z}$ (BALLÉ et al., 2018; QIAN et al., 2022). Os parâmetros de convolução são indicados como: número de filtros $\times$ altura de suporte do *kernel* $\times$ largura de suporte do *kernel* / *stride* para *down* ou *upsampling*, onde $\uparrow$ indica *upsampling* e $\downarrow$ *downsampling* - Imagem Retirada e Adaptada de (BALLÉ et al., 2018)

4.2 Arquitetura Autoencoder Variacional com Hyperprior

A presente subseção foi escrita com base nos trabalhos de: (BALLÉ et al., 2018; MINNEN; BALLÉ; TODERICI, 2018; OLIVEIRA et al., 2021; QIAN et al., 2022).

Os *autoencoders* foram elaborados com o intuito de auxiliar em tarefas de redução de dimensão de dados semelhantes. No contexto da compressão de imagem, os *autoencoders* são usados para aprender uma representação com baixa entropia, ou seja, pouca informação e padrões repetitivos, após o processo de quantização. Para entender melhor o processo de quantização e codificação aritmética, pode-se pensar neste processo unidimensionalmente. A ideia da rede neural é transformar a imagem, forçando repetições na forma como ela é representada, onde a imagem é convertida para uma grande *string* composta por muitos símbolos, sendo cada símbolo um conjunto de *floats*. O processo de quantização, necessário antes da aplicação da codificação aritmética, é responsável por arredondar os valores *float*, os transformando de contínuos para discretos. O processo de compressão está relacionado com a repetição de valores discretos, visto que dificilmente um valor contínuo se repete, por essa razão é preciso aplicar a quantização para que haja a conversão. Após a conversão, o codificador aritmético, que é um algoritmo que usa um número baixo de *bits* para compactar dados, verifica quantas vezes determinado padrão de símbolo se repete. Se a probabilidade de ocorrência de determinado símbolo é alta, o codificador aritmético vai usar menos bits para codificar a informação, e caso a probabilidade seja baixa, mais *bits* serão utilizados para a codificação (LANGDON, 1984; SALMAN, 2016; AHMED GAD, 2021).

Basicamente, quando aplicada a compressão, a arquitetura *autoencoder* é composta por uma Transformação de Análise g_a e uma Transformação de Síntese g_s conectada por um gargalo que realiza o processo de quantização e codificação (Figura 12 - destaque em vermelho). Um *autoencoder* auxiliar também é usado para inferir a distribuição de probabilidade da representação latente (OLIVEIRA et al., 2021) como apresentado pelo destaque verde na Figura 12 (BALLÉ et al., 2018). Essa arquitetura auxiliar é o *Hyperprior*. Através dele, é possível descobrir as estatísticas da imagem, que por sua vez auxiliam a aplicação da codificação/decodificação aritmética. Quando se conhece as estatísticas da imagem, é possível comprimir melhor a informação, pois como foi apresentado, a compressão é baseada em padrões espaciais que se repetem. Assim, o objetivo é possuir uma representação que seja fácil para a codificação/decodificação aritmética comprimir, pois conhecendo as distribuições, a compressão ocorre de maneira mais eficaz.

A arquitetura da Figura 12 é um refinamento do *autoencoder* tradicional, o que leva a um maior desempenho às custas de uma maior complexidade, por meio de uma melhor adaptação à imagem de entrada (OLIVEIRA et al., 2021). Segundo (OLIVEIRA et al., 2021), o *autoencoder* variacional permite atingir desempenho de compressão de

última geração, próximo ao formato de arquivo para codificação de imagens digitais chamado BPG (*Better Portable Graphics*) às custas de um considerável aumento de complexidade em relação a arquitetura de *autoencoder* mais simples e tradicional apresentado em (BALLÉ; LAPARRA; SIMONCELLI, 2016c).

No *autoencoder* principal, destacado em vermelho, na Figura 12, a transformação de análise g_a é aplicada aos dados de entrada x para produzir uma representação $y = g_a(x)$. Após o gargalo, a transformada de síntese g_s é aplicada à representação quantizada $\hat{y}$ para reconstruir a imagem $\hat{x} = g_s(\hat{y})$. Tanto a transformação de análise g_a , como o transformada de síntese g_s são provenientes de três camadas convolucionais, cada uma constituída por N filtros com *kernel* $n \times n$ relativo a funções de ativação paramétrica chamadas GDN e IGDN, como já explicado na subseção 3.1, e *downsampling* e *upsampling* por um fator 2. O *downsampling* reduz a dimensionalidade da imagem utilizando etapas de *pooling*, compactando a informação e extraíndo apenas as informações consideradas úteis em uma determinada região (ANJOS, 2019). Já o *upsampling* restabelece a imagem para a sua resolução original (SKEIKA et al., 2019).

Essas três camadas de convolução estão ligadas à entrada do gargalo por uma camada de convolução composta por $M > N$ filtros com o mesmo *kernel*, mas sem função de ativação. Ao contrário das funções usuais de ativação sem parâmetros, por exemplo ReLU, *softmax*, *sigmoid* e outras, GDN e IGDN são funções paramétricas que implementam uma normalização adaptativa (OLIVEIRA et al., 2021). Segundo (BALLÉ, 2018; OLIVEIRA et al., 2021), apesar de o GDN/IGDN ser mais complexo, quando comparado com as funções de ativação comumente utilizadas, ele aumenta o desempenho da compactação.

A *interface* entre a transformada de análise e a transformada de síntese, o chamado gargalo, é composta por um quantizador que produz o vetor de valor discreto $\hat{y} = Q(y)$, um codificador de entropia, baseado nos padrões que se repetem ou não, e o decodificador associado. O processo de desquantização é feito pela transformação de síntese g_s (e por h_s no caso do *autoencoder* variacional). Um método de codificação de entropia padrão, como aritmética ou codificação de *Huffman*, compacta sem perdas a representação de dados quantizados, explorando sua distribuição estatística. Os parâmetros do *autoencoder* são aprendidos conjuntamente através da otimização da função *loss* ($\mathcal{L}$) envolvendo a taxa de *bits* $R(\hat{y})$ e a distorção $D(x, \hat{x})$ entre a imagem original x e a imagem reconstruída $\hat{x}$. O critério de taxa de distorção, denotado como $J(x, \hat{x}, \hat{y})$, escreve como a soma ponderada:

$$J(x, \hat{x}, \hat{y}) = \lambda D(x, \hat{x}) + R(\hat{y}) \rightarrow \mathcal{L} = \lambda D + R, \quad (5)$$

onde λ é um parâmetro chave que ajusta a compensação taxa-distorção. Neste contexto, por possuir essa função, quanto maior o valor de λ mais importância está sendo

atribuída a distorção, ou seja, a imagem resultante na saída é mais similar, em qualidade visual, com a imagem de entrada, enquanto que a compressão aplicada é baixa. Porém, quando o valor de λ é pequeno, menos importância está sendo atribuída a distorção, ou seja, a imagem resultante na saída é menos similar, em qualidade visual, com a imagem de entrada, mas a compressão aplicada é alta. Na seção 6.3.4 é mostrada uma avaliação que exemplifica esse ajuste proporcionado pelo parâmetro λ tanto na distorção (D), como na taxa (R).

A taxa de *bits* R alcançada por um codificador de entropia é limitada pela entropia derivada da distribuição de probabilidade discreta real $m(\hat{y})$ do vetor quantizado $\hat{y}$. O aumento da taxa vem da incompatibilidade entre o modelo de probabilidade $p_{\hat{y}}(\hat{y})$ necessário para o projeto do codificador e $m(\hat{y})$. A taxa de *bits* é dada pela entropia cruzada de *Shannon* entre as duas distribuições:

$$H(\hat{y}) = E_{\hat{y} \sim m} [-\log_2 p_{\hat{y}}(\hat{y})]. \quad (6)$$

A taxa de *bits* é então minimizada se o modelo de distribuição $p_{\hat{y}}(\hat{y})$ for igual à distribuição $m(\hat{y})$ decorrente da distribuição real da imagem de entrada e da transformação de análise g_a . Isso destaca o papel fundamental do modelo de probabilidade. A medida de distorção D é escolhida para levar em conta a qualidade da imagem percebida por um observador humano. Devido às suas muitas propriedades computacionais desejáveis, o Erro Quadrático Médio (MSE) é geralmente selecionado.

Um elemento importante nas estruturas de compressão de imagens aprendidas de ponta a ponta (*end-to-end*) é o modelo de entropia definido através do modelo de probabilidade $p_{\hat{y}}(\hat{y})$, que prediz a distribuição de probabilidade da representação latente quantizada nos módulos de codificação e decodificação. Ele estima a distribuição de probabilidade condicional de latentes para compressão por algoritmos de codificação de entropia padrão. Um dos principais desafios do modelo de entropia é aprender a representar as dependências da imagem, como: textura, semântica e espaço de cor (QIAN et al., 2022). Para superar esses desafios e obter um modelo de entropia mais realista e mais adaptativo, (BALLÉ et al., 2018) propôs um modelo *Hyperprior*, derivado de um *autoencoder* variacional, que leva em consideração cada imagem de entrada (OLIVEIRA et al., 2021).

Dentre os tipos de modelos de entropia, o *Hyperprior* (MINNEN et al., 2018; BALLÉ et al., 2018; QIAN et al., 2022) utiliza as informações secundárias da representação latente quantizada com *bits* adicionais. Variáveis aleatórias auxiliares $\hat{z}$ são derivadas de y pelo *autoencoder* auxiliar (*Hyperprior*), conectado em paralelo com o gargalo (Figura 12 - destaque em verde). Em primeiro lugar, a transformada de análise do *hyperprior* h_a produz o conjunto de variáveis aleatórias auxiliares z . O z é então quantizado, comprimido e transmitido como informação secundária. O *Arithmetic Encoder/Decoder* do *autoencoder* destacado em vermelho (Figura 12) utiliza as informações resultantes da

saída do h_s , tanto para comprimir, como para descomprimir. Supondo que uma imagem precise ser enviada para um dispositivo, é necessário enviar o espaço latente comprimido e o *hyperprior* comprimido, pois é preciso primeiramente descomprimir o *hyperprior* e com essa informação, descomprimir o espaço latente do *autoencoder* principal. Nesse caso, denomina-se informação secundária, pois a imagem comprimida será composta também por essa informação resultante do h_s , que é pequena, mas que ajuda na descompressão da imagem. Pode-se entender a informação secundária como uma chave para a descompactação da imagem. Em segundo lugar, z é modificado pela transformação de síntese do *hyperprior* h_s em um segundo conjunto de variáveis aleatórias σ . A distribuição z é assumida totalmente fatorada e cada elemento de representação, conhecendo z , é modelado por uma distribuição Gaussiana de média zero com seu próprio desvio padrão σ .

O objetivo do treinamento para compressão de imagem aprendida é otimizar a compensação entre o comprimento de codificação estimado do fluxo de *bits* e a qualidade da reconstrução, que é um problema de otimização de distorção de taxa (QIAN et al., 2022):

$$\mathcal{L} = R + \lambda D = \underbrace{\mathbb{E}_{x \sim p_x} [-\log_2 p_{\hat{y}}(\hat{y})]}_{\text{taxa (latentes)}} + \underbrace{\mathbb{E}_{x \sim p_x} [-\log_2 p_{\hat{z}}(\hat{z})]}_{\text{taxa (hyper-latentes)}} + \lambda \cdot \underbrace{\mathbb{E}_{x \sim p_x} \|x - \hat{x}\|}_{\text{distorção}}, \quad (7)$$

onde p_x é a distribuição de imagens e p_z usa um modelo de densidade totalmente fatorado. O primeiro termo representa a taxa de compressão estimada da representação latente, enquanto o segundo termo representa a taxa de compressão estimada da representação hyper-latente. O terceiro termo representa o valor de distorção sob a métrica de erro quadrático médio (MSE). A Tabela 2 apresenta um resumo dos símbolos utilizados.

Tabela 2: Resumo dos Símbolos

Símbolo	Significado
x	Imagem Original
$\hat{x}$	Imagem Reconstruída
y	Representação Latente Aprendida
$\hat{y}$	Representação Latente Aprendida Quantizada
z	Representação Hyper-Latente
$\hat{z}$	Representação Hyper-Latente Quantizado
$g_a(x)$	Transformação de Análise
$g_s(\hat{y})$	Transformação de Síntese
$h_a(y)$	Transformação de Análise <i>Hyperprior</i>
$g_s(\hat{z})$	Transformação de Síntese <i>Hyperprior</i>
$Q(y)$	Quantizador
λ	Ajusta a Compensação Taxa-Distorção
$\hat{\sigma}$	Desvio Padrão de uma Distribuição Gaussiana de Média Zero
$R(\hat{y})$	Taxa de <i>Bits</i>
$D(x, \hat{x})$	Distorção entre a Imagem Original x e a Imagem Reconstruída $\hat{x}$

5 USO DO DESCRITOR SIFT NA QUANTIFICAÇÃO DA INFORMAÇÃO

O presente capítulo tem o intuito de introduzir e explicar o uso do SIFT, bem como apresentá-lo como uma métrica baseada em descritores para quantificar a variação e diversidade de informação em uma imagem, aqui denominada Quantidade de SIFT (QS).

5.1 Introdução ao Scale-Invariant Feature Transform (SIFT)

O SIFT é um dos algoritmos utilizados para a extração de características, que por definição é todo um conjunto de operações de processamento e análise de imagens que são realizadas com o objetivo de obter valores numéricos que caracterizem e descrevam uma determinada imagem ou parte dela (LOWE, 2004a, 2004b; DREWS, 2008; OLIVEIRA, 2013; JESUS; ROCHA; BITTENCOURT, 2019; SZELISKI, 2022). Neste contexto, o SIFT é um descritor usado para extrair pontos característicos salientes em uma imagem. Para cada um desses pontos, ele também fornece um conjunto de características que descrevem uma pequena região da imagem ao redor do ponto, sendo que esses recursos são invariáveis à rotação e escala da imagem (DREWS, 2008; KABBAI et al., 2015; JESUS; ROCHA; BITTENCOURT, 2019). Conforme (BOTAN, 2016; SZELISKI, 2022) o processo genérico de detecção e extração de características de uma imagem pode ser dividido em dois passos, descritos a seguir.

5.1.1 Localização de Patches e Keypoints

O *patch* é conhecido como a região em torno de um *keypoint*, e de maneira básica este último é encontrado em regiões com grandes contrastes e/ou com uma mudança direcional de intensidade ou cor, ou seja, regiões que apresentam gradientes de valores mais elevados (BOTAN, 2016). A Figura 13¹ mostra regiões para onde os gradientes crescem.

¹ A seta indica o gradiente, a cor preta significa que o valor real da função é alto e quanto mais branco o valor, menor o valor da função. Em outras palavras, o gradiente mostra uma direção positiva à medida que o valor da função vai de pequeno para grande, ou seja, o vetor aponta para a maior mudança de intensidade ou cor.

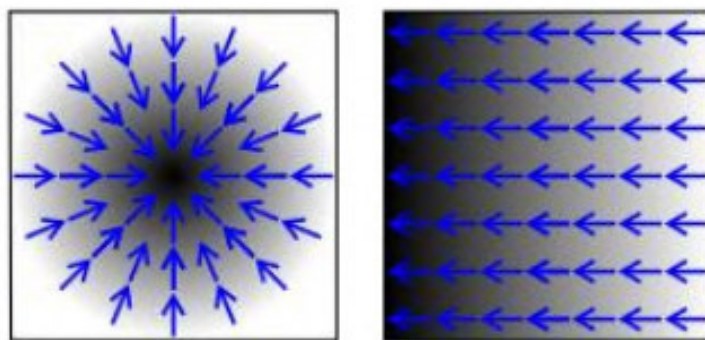


Figura 13: Análise do Gradiente - Retirado de (BOTAN, 2016)

O cálculo vetorial define o gradiente de um campo escalar como sendo um campo de vetores que apontam na direção da maior taxa de aumento da grandeza escalar, e cuja grandeza é a da maior taxa de variação. Na Figura 13 o campo escalar está representado por tons de cinza, sendo as regiões mais escuras de valores mais altos. Os valores de gradiente correspondentes estão representados pelas setas azuis.

5.1.2 Extração de Descritores

Nesta etapa os *patches* já extraídos são convertidos e agrupados em formas mais compactas, estáveis e invariantes, ou seja, são transformados em descritores. Este processo consiste basicamente em orientar os *patches*, rotacioná-los e escaloná-los. A Figura 14 ilustra melhor como é esse processo. É importante destacar que esses descritores devem ser distintos, robustos a ruídos, a detecção de erros e deformações, tanto geométricas quanto fotométricas (distorções de lentes) (BOTAN, 2016).

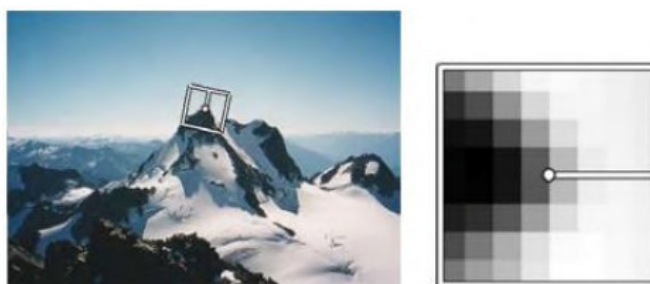


Figura 14: Extração e Orientação de um Descritor - Retirado de (BROWN; SZELISKI; WINDER, 2005; BOTAN, 2016)

O SIFT utiliza uma aproximação do Laplaciano do Gaussiano para encontrar os *key-points*; o Laplaciano para encontrar regiões de mudanças bruscas de gradiente (detector de bordas), e o Gaussiano como filtro (para remover ruído), já que o Laplaciano é sensível a ruídos (BOTAN, 2016). No SIFT, o processo de descrição das *features* é feito considerando uma vizinhança de 16×16 no entorno do *keypoint*. O passo seguinte é dividi-la em

16 sub-regiões com 4×4 pixels e para cada sub-região um vetor de 128 dimensões para cada *keypoint* é criado (BOTAN, 2016).

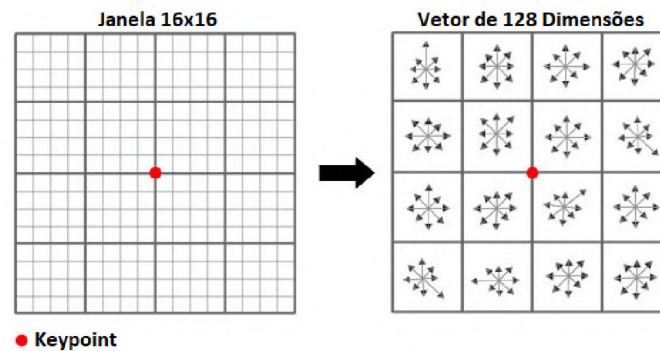


Figura 15: Descritor de *Keypoint* - Retirado de (TYAGI, 2019)

5.2 Aplicação do SIFT na Quantificação da Informação

Com o intuito de verificar a ação e o comportamento do SIFT na quantificação da diversidade de informação presente na imagem e fundamentar a ideia do descritor, avaliou-se o número de *features* retornado pelo algoritmo SIFT (*QS*) frente a uma série de imagens não subaquáticas e subaquáticas. A Figura 16 apresenta um gráfico de pontos, onde avalia-se a quantidade de descritores em relação ao quão monótona é cada imagem. Percebe-se mais a esquerda imagens com predominância de fundo (monótonas), enquanto que mais a direita tem-se imagens com mais objetos e feições (menos monótonas).

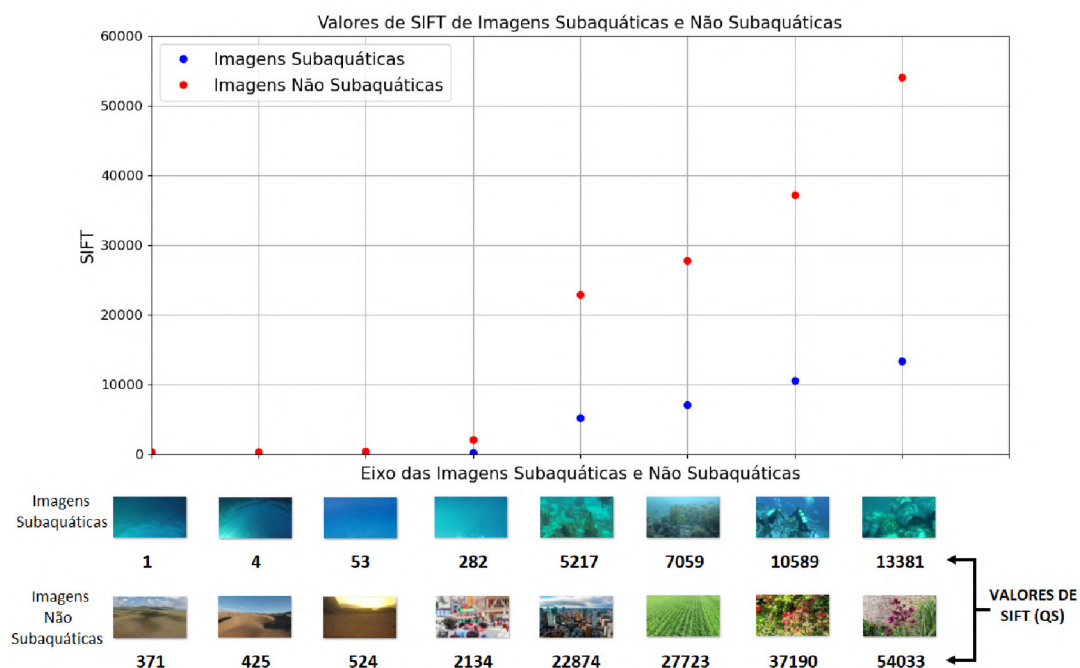


Figura 16: Valores de SIFT de Imagens Subaquáticas e Não Subaquáticas

É possível perceber visualmente que as imagens que são mais monótonas/homogêneas possuem um valor menor de SIFT, enquanto que imagens com mais detalhes e textura possuem um valor maior de SIFT. Os pontos vermelhos que representam as imagens não subaquáticas tendem a um maior crescimento de valor de SIFT do que os pontos azuis, que representam as imagens subaquáticas, isso porque a variação do valor SIFT resultante é muito maior nas imagens não subaquáticas, que apresentam mais informação. Através desta análise, pode-se levantar a hipótese de que após um certo número de características (valor de SIFT), não há mais significado e informações relevantes na imagem, ou seja, a grande quantidade de pontos resultantes do algoritmo se deve a textura ou grande quantidade de detalhes presentes na imagem, por exemplo. Por isso, para não afetar o desempenho de compressão da função de perda ($\mathcal{L}$) e facilitar a convergência do algoritmo de aprendizado foi colocado um limiar, nos valores de SIFT. Foi escolhido o valor de 1000, pois sem o limiar, devido aos altos valores de SIFT resultantes, haveria uma grande tendência de o treinamento não aprender a comprimir. Neste contexto, em resumo, para imagens com um certo nível de complexidade não deseja-se alterar a compressão. O foco será em imagens com baixa complexidade, por isso, define-se um limiar.

Assim, propõe-se um método que fará uso do número de descritores retornados pelo SIFT para comprimir mais ou menos a imagem, servindo como métrica para estimativa de informação presente na imagem. Além disso, propõe-se o uso de uma abordagem de Normalização Min-Max apresentada em (HENDERI; WAHYUNINGSIH; RAHWANTO, 2021) e definida por:

$$X_{novo} = \frac{X - X_{min}}{X_{max} - X_{min}}. \quad (8)$$

onde X_{novo} é o novo valor de SIFT normalizado, X é o valor antigo, X_{min} é o valor mínimo de SIFT no *dataset* e X_{max} é o valor máximo de SIFT no *dataset*. Como já citado, o objetivo da otimização taxa-distorção é que, ao calcular a função de perda (*loss*), a distorção (D) e a taxa (R) devem garantir baixa distorção e baixa taxa de *bits*, o que é mais útil ao transmitir uma informação (BALLÉ et al., 2018; OLIVEIRA et al., 2021). Imagens subaquáticas possuem características próprias, sendo assim tanto a distorção como a taxa de *bits* podem ser trabalhadas, para esse conjunto de dados, levando em conta a quantidade de informação e características presentes na imagem. Desta forma a equação (5) pode ser redefinida por:

$$\mathcal{L} = \lambda * (QS^2 + 0,001) * D + R, \quad (9)$$

onde QS é uma medida da quantidade de SIFT estimada, ou seja, a quantidade de informação presente na imagem resultante da leitura do SIFT, que ao quadrado expande a métrica, penalizando a distorção. A soma de 0,001 garante que haja distorção mínima, ou seja, garante que a imagem reconstruída tenha uma qualidade mínima.

Nos testes realizados neste trabalho, a função *Loss* ($\mathcal{L}$), descrita pela equação (9) foi utilizada somente para o treinamento apresentado na subseção 6.3.5, onde o SIFT foi incluso.

A Figura 17 mostra o relacionamento entre a categoria da imagem (as categorias são explicadas em detalhes na Tabela 3) e o número de *features* retornado pelo SIFT. Como era de se esperar, imagens subaquáticas monótonas (linha azul escuro) apresentam baixos valores de SIFT, bem como imagens "meio termo", onde mesmo com a presença de objetos na cena, muitas regiões da imagem ainda são compostas de água, com poucas *features* destacadas. O uso do SIFT parece ser realmente adequado para descrição do nível de monotonia nas imagens. Ainda, ao observar o gráfico da imagem subaquática, composta por cenários povoados de objetos (linha verde), nota-se altos números de *features* retornados, semelhantes a imagens *indoor* e *outdoor*. Estas avaliações apontam como adequada a escolha de uso do SIFT como informação para parametrização de processo de compressão.

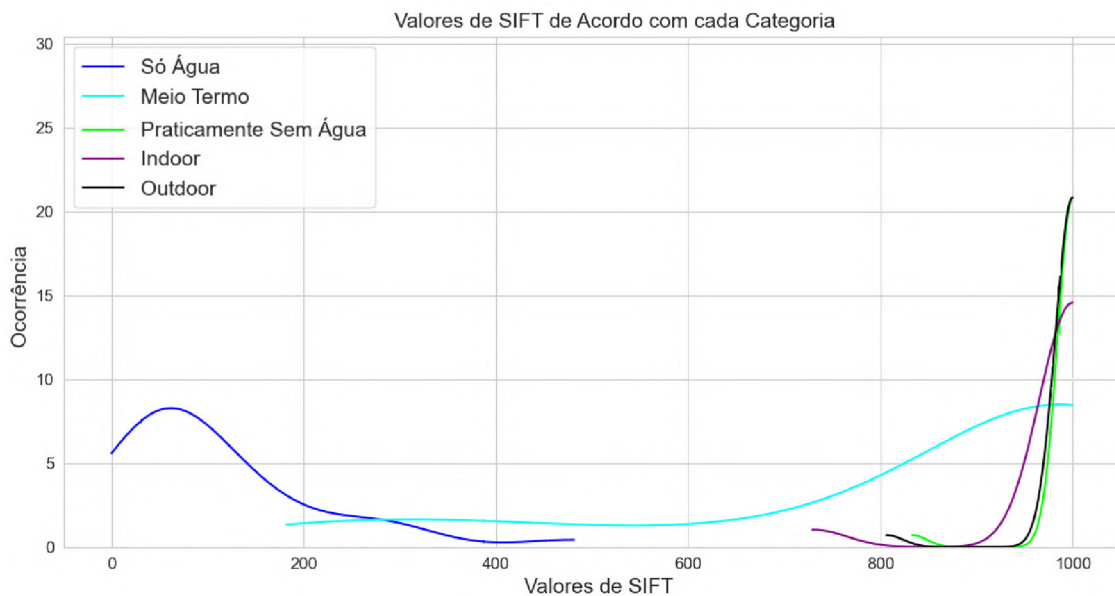


Figura 17: Valores de SIFT de Acordo com cada Categoria

6 EXPERIMENTOS, TESTES E RESULTADOS

Na presente seção são apresentados e discutidos os experimentos e resultados obtidos a partir do método demonstrado na subseção 4.2, e também em conjunto com o SIFT como métrica para quantificação da informação da imagem, bem como uma metodologia para obtenção dos *datasets* de imagem subaquática a serem utilizados no trabalho.

6.1 Plataforma CompressAI

A plataforma *CompressAI* é uma biblioteca *PyTorch* para avaliação e pesquisa de compressão (BÉGAINT et al., 2020). Uma vez que o campo de pesquisa da compressão de dados utilizando aprendizado profundo é relativamente recente e faltam ferramentas que auxiliem e facilitem as contribuições de pesquisadores da área, a plataforma foi desenvolvida com o intuito de melhorar essa situação. O *CompressAI* reproduz resultados da literatura e permite que pesquisadores façam modificações, treinem e avaliem seus próprios *codecs* baseados em redes neurais. Por meio da plataforma é possível desenvolver um *pipeline* de pesquisa completo, do treinamento à avaliação de desempenho em relação a *codecs* de aprendizado e convencionais (BÉGAINT et al., 2020), disponibilizando operações personalizadas, modelos e ferramentas para pesquisar, desenvolver e avaliar *codecs* de compressão de imagem. A plataforma também inclui modelos pré-treinados (usando o conjunto de dados de treinamento Vimeo-90K (BÉGAINT et al., 2020)) e ferramentas de avaliação para comparar métodos aprendidos com *codecs* tradicionais. A plataforma é disponibilizada no *GitHub*¹, onde sofre constantes atualizações e modificações (BÉGAINT et al., 2020).

Os *codecs* baseados em redes neurais artificiais são capazes de igualar ou superar as abordagens convencionais. O objetivo do treinamento é minimizar o comprimento estimado do fluxo de *bits*, mantendo a distorção da imagem reconstruída baixa, em comparação com o conteúdo original. A distorção pode ser facilmente medida com métricas objetivas ou perceptuais como o MSE (erro quadrático médio) ou o MS-SSIM (similaridade estrutural multi-escala) (BÉGAINT et al., 2020). Múltiplas arquiteturas do

¹<https://github.com/InterDigitalInc/CompressAI>

estado da arte em compressão de imagem (BALLÉ et al., 2018; MINNEN; BALLÉ; TORDERICI, 2018; CHENG et al., 2020) foram reimplementadas no *PyTorch* com camadas e módulos específicos fornecidos pelo *CompressAI*. É comum no *PyTorch* que modelos pré-treinados recebam lotes de entrada de tensores de imagem RGB de forma $(N, 3, H, W)$, com N sendo o *batch size* e (H, W) as dimensões espaciais das imagens.

O número de canais no espaço latente do *autoencoder* varia de acordo com as taxas de *bits* desejadas. Assim sendo, o número de canais precisa ser maior para taxas de *bits* mais altas. Para taxas de *bits* baixas, abaixo de 0,5 bpp (*bit por pixel*), a literatura geralmente recomenda o uso de 192 canais para o espaço latente e 320 canais para taxas de *bits* mais altas (BALLÉ et al., 2018; BÉGAINT et al., 2020). Como salientado, a plataforma *CompressAI* possui vários modelos estado da arte em compressão de imagens. Para o presente trabalho foi escolhido o modelo de *Autoencoder* Variacional com *Hyperprior*, que foi apresentado e explicado na subseção 4.2.

6.2 Geração de Dataset de Imagem Subaquática

O treinamento da rede *deep learning* necessita de um *dataset* de imagens capaz de representar situações de missões subaquáticas reais. Para a realização de testes com o modelo de rede *autoencoder* variacional foi realizada uma curadoria que resultasse em cenários de operações de AUVs. Devido a dificuldade de se encontrar um grande número de imagens subaquáticas, optou-se por pesquisar por vídeos onde mergulhadores filmam suas atividades de mergulho, disponíveis na internet, e retirar os *frames* dos mesmos para criar o *dataset* necessário.

Neste contexto, para criação do conjunto de dados de imagens subaquáticas foram utilizados alguns vídeos, onde três deles são destacados somente para exemplificação²³⁴. Após a coleta dos *frames* foram selecionadas 9695 imagens com dimensão de 1920×1080 , sendo que 7756 imagens foram destinadas ao treinamento (80%) e 1939 imagens para o teste (20%). A Figura 18 apresenta uma pequena amostra das imagens presentes em ambos os *datasets*, apenas para exemplificação.

²Vídeo 1 - Canal: Biodun Onipede (Link: https://www.youtube.com/watch?v=7FTNmlbYn-ot=105sab_channel=BiodunOnipede)

³Vídeo 2 - Canal: TFIL (Link: https://www.youtube.com/watch?v=v8eSPf4RzTUab_channel=TFIL)

⁴Vídeo 3 - Canal: Axel Hein (Link: https://www.youtube.com/watch?v=id47Nla9ZToab_channel=AxelHein)

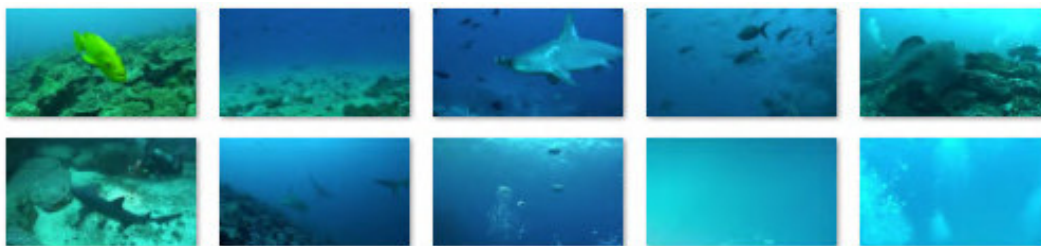
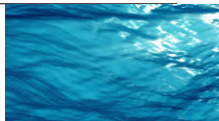
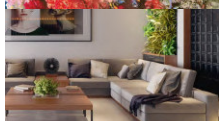


Figura 18: Amostra das Imagens Subaquáticas Presentes nos *Datasets* de Treino e Teste

Também foi criado um *dataset* de validação para avaliar o modelo proposto, dividido em cinco categorias de imagem, associadas as características da cena, onde cada possui 30 imagens. Para tal, foram retiradas imagens aleatórias da internet que atendessem as categorias determinadas, que são: *Só Água*, *Meio Termo*, *Praticamente Sem Água*, *Indoor* e *Outdoor*. A Tabela 3 apresenta uma amostra e explicação do que representam as cinco categorias especificadas.

Tabela 3: Amostra do *Dataset* de Validação

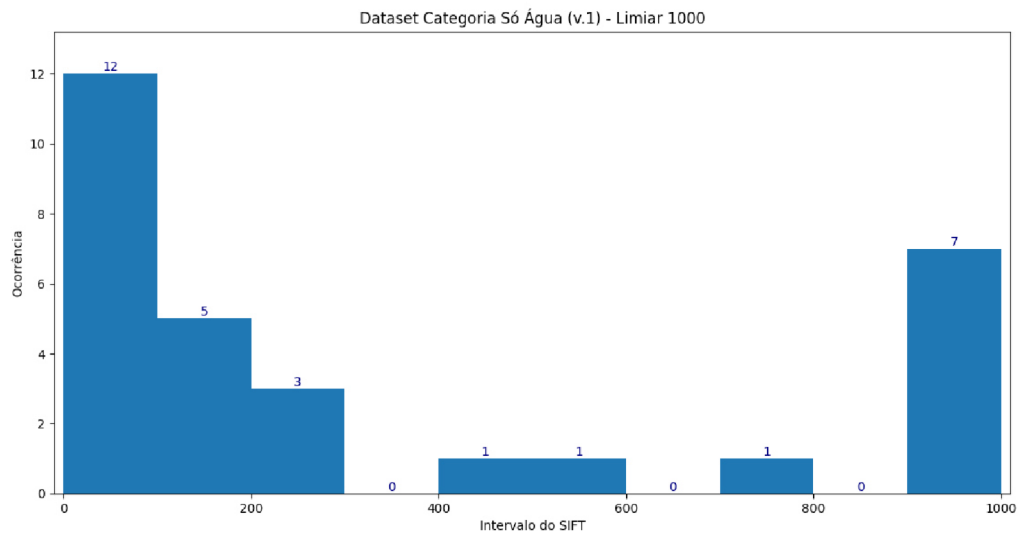
Categoria	Explicação	Imagem
Só Água	-Imagens monótonas subaquáticas, sem objetos presente, apenas fundo d'água.	
Meio Termo	-Imagens subaquáticas que apresentam cenários com a presença de algum objeto, mas com predominância de fundo d'água.	
Praticamente Sem Água	-Imagens subaquáticas mas com foco em algum objeto ou detalhe da cena, onde a visualização dos objetos é mais relevante.	
Indoor	-Imagens em ambiente interno, por exemplo, ambientes domiciliares ou comerciais.	
Outdoor	-Imagens em ambiente externo, por exemplo, natureza.	

6.3 Resultados Experimentais

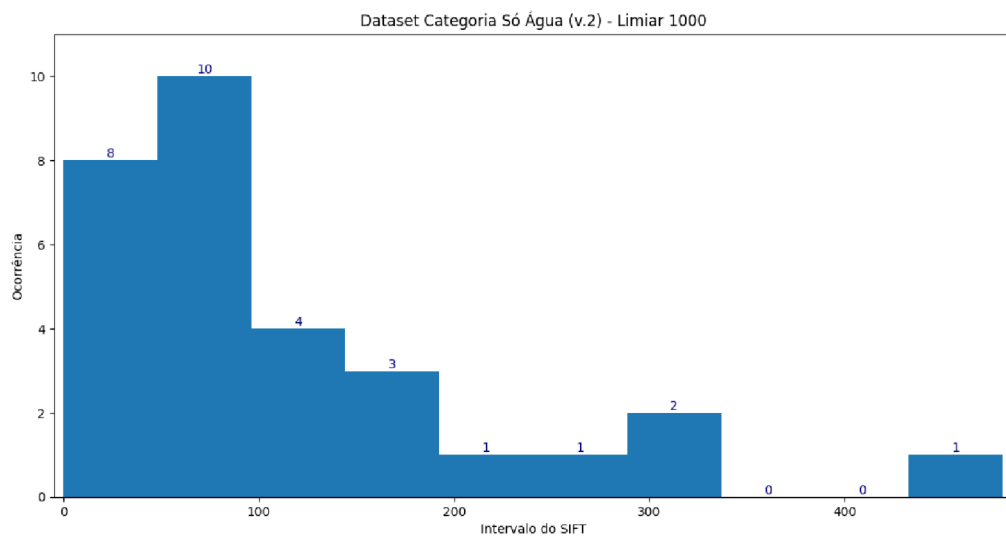
Com o objetivo de se avaliar, em um primeiro momento, o desempenho do método *Autoencoder Variacional com Hyperprior* e logo após com o uso da medida de quantidade de SIFT (QS), nesta subseção são apresentados testes e discutidos resultados obtidos frente a diferentes contextos. O primeiro, na subseção 6.3.1, avalia a inferência do *dataset subaquático* em conjunto ao modelo *autoencoder variacional com hyperprior* pré-treinado com *dataset* de imagens não subaquáticas. O segundo, na subseção 6.3.2, avalia o modelo que nesse caso foi totalmente treinado com *dataset* de imagens subaquáticas. O terceiro, na subseção 6.3.3, aplica um ajuste fino ao modelo original, com base no *dataset subaquático*. Na subseção 6.3.4, a rede *Autoencoder Variacional com Hyperprior* é treinada, avaliando-se o parâmetro λ na taxa de compressão e na qualidade visual da imagem. Por fim, a subseção 6.3.5 analisa o uso do SIFT como uma métrica para quantificar a variação de informação em uma imagem, como explicado na subseção 5.2.

Porém, é importante destacar alguns pontos a respeito dos testes aqui apresentados:

- Os testes apresentados nas subseções: 6.3.1, 6.3.2, 6.3.3 e 6.3.4 foram avaliados dentro da Categoria *Só Água*, por imagens com a distribuição em relação ao valor de SIFT apresentada na Figura 19(a) e com a função de perda dada pela equação (5);
- Os testes apresentados nas subseções: 6.3.2 e 6.3.3 foram treinados com imagens com a distribuição em relação ao valor de SIFT apresentada na Figura 20(a);
- O teste apresentado na subseção: 6.3.5 foi avaliado dentro da Categoria *Só Água*, por imagens com a distribuição em relação ao valor de SIFT apresentada na Figura 19(b) e com a função de perda dada pela equação (9);
- O teste apresentado na subseção: 6.3.5 foi treinado com imagens com a distribuição em relação ao valor de SIFT apresentada na Figura 20(b).

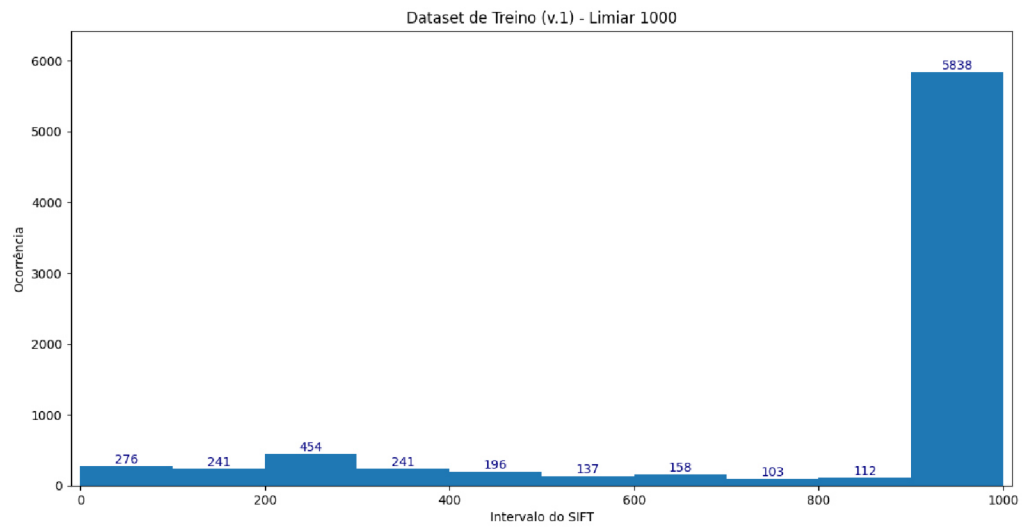
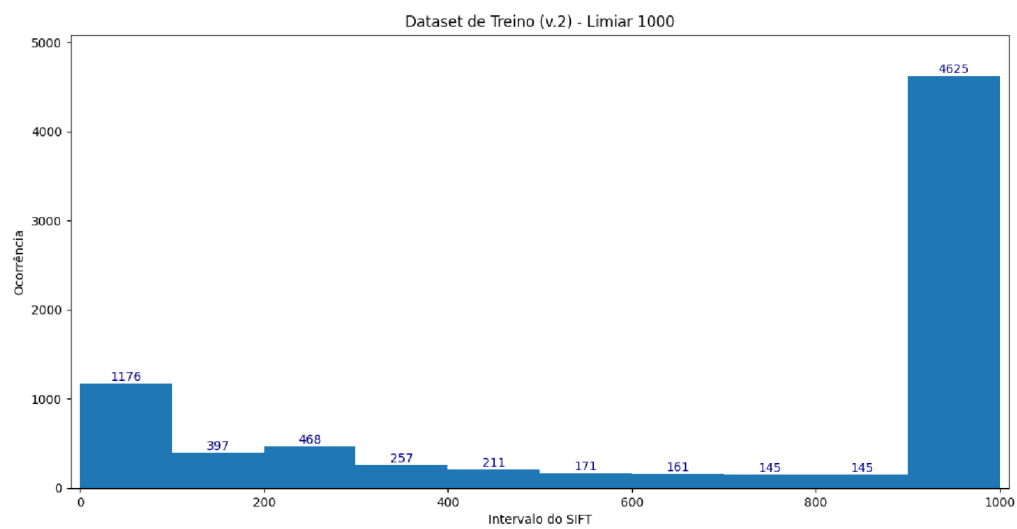


(a) Primeira Versão do *Dataset* da Categoria *Só Água*

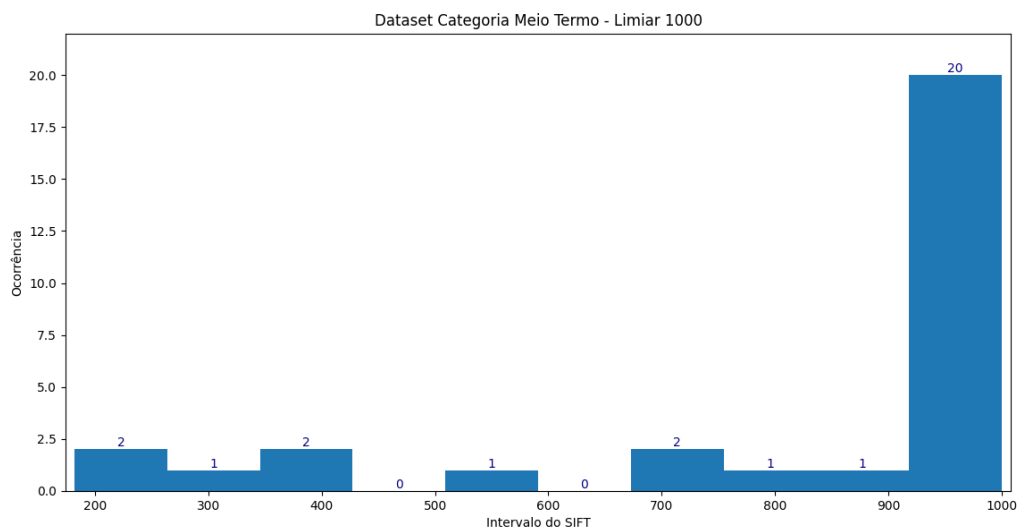


(b) Segunda Versão do *Dataset* da Categoria *Só Água*

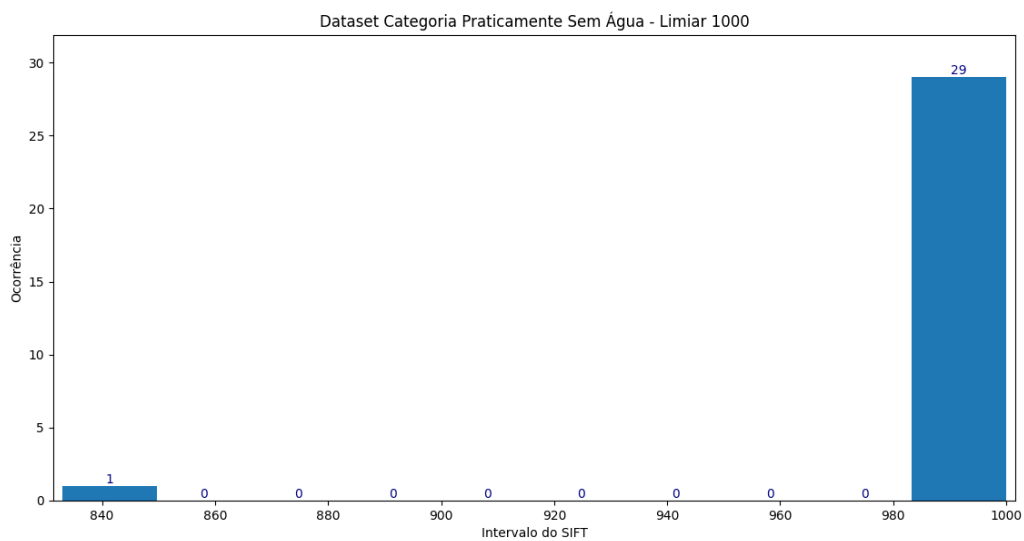
Figura 19: Distribuição *Dataset*: Categoria *Só Água*

(a) Primeira Versão do *Dataset* de Treinamento(b) Segunda Versão do *Dataset* de TreinamentoFigura 20: Distribuição *Dataset*: Treinamento

Somente pra visualização, as Figuras 21(a), 21(b), 22(a) e 22(b) mostram respectivamente a distribuição de valor de SIFT das categorias: *Meio Termo*, *Praticamente Sem Água*, *Indoor* e *Outdoor*, que foram utilizadas na avaliação de todos os testes.

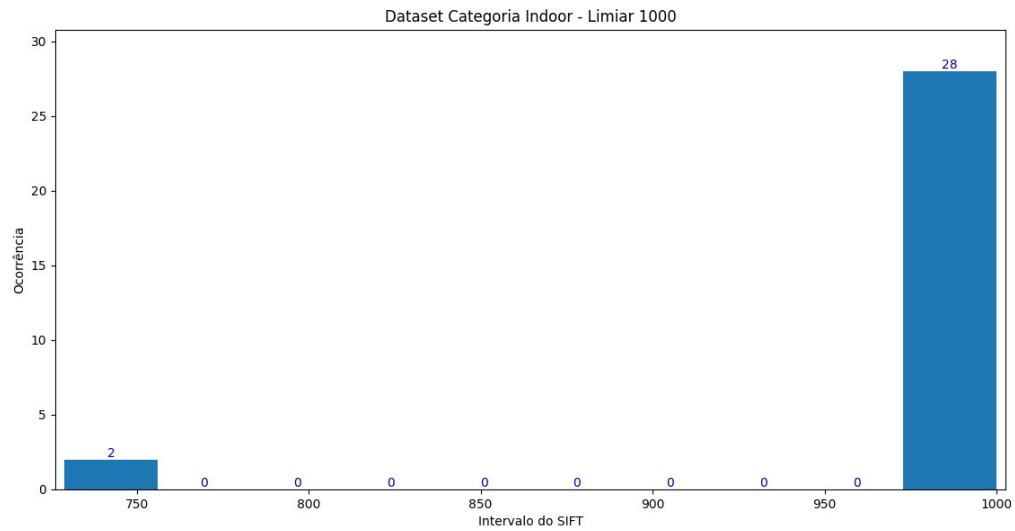
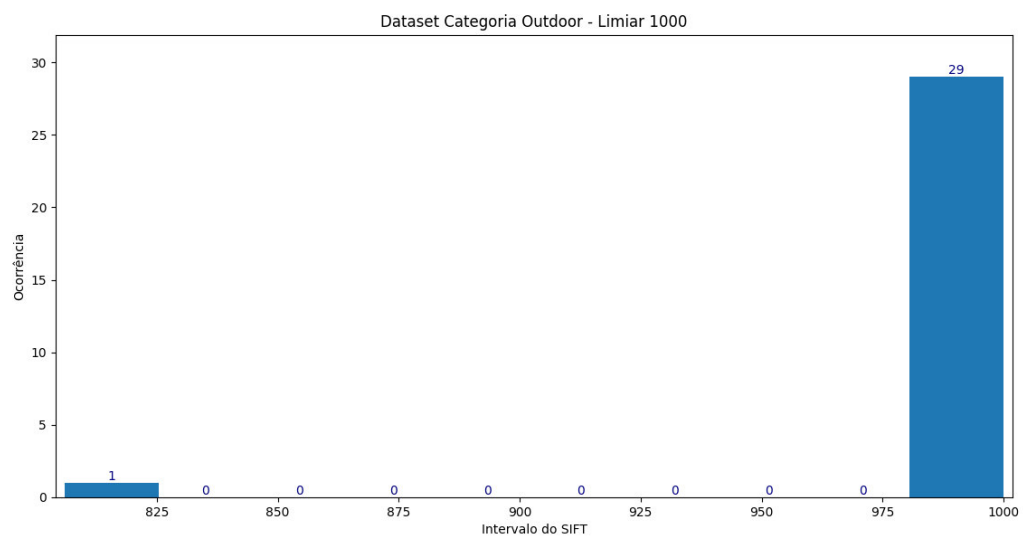


(a) Categoria *Meio Termo*



(b) Categoria *Praticamente Sem Água*

Figura 21: Distribuição *Dataset*: Categorias *Meio Termo* e *Praticamente Sem Água*

(a) Categoria *Indoor*(b) Categoria *Outdoor*Figura 22: Distribuição *Dataset*: Categorias *Indoor* e *Outdoor*

6.3.1 Avaliação da Inferência da Rede Original Autoencoder Variacional com Hyperprior em Imagens Subaquáticas (Original)

Estes testes buscam avaliar a inferência do *dataset subaquático* junto à rede *autoencoder* variacional pré-treinada com *dataset* não subaquático. Esta avaliação servirá como *benchmark* para determinar se os demais experimentos, apresentados nas subseções 6.3.2 e 6.3.3 estão contribuindo ou não com o objetivo de compressão de imagens subaquáticas. Os valores de PSNR e BPP obtidos podem ser visualizados na Tabela 5.

6.3.2 Rede Autoencoder Variacional com Hyperprior Treinada com Imagens Subaquáticas (Totalmente Treinado)

Testes foram realizados com o modelo *Autoencoder* Variacional com *Hyperprior* totalmente treinado com o *dataset subaquático* proposto na subseção 6.2. A Tabela 4 apresenta os parâmetros utilizados para o treinamento. A placa de vídeo utilizada em todos os treinamentos foi o modelo GeForce GTX TITAN X.

Tabela 4: Parâmetros de Treinamento - *Autoencoder* Variacional com *Hyperprior* Treinado com Imagens Subaquáticas

Parâmetro	Valor
Número de Épocas	200
Imagens de Treino (80%)	7756
Imagens de Teste (20%)	1939
<i>Batch Size</i>	16
<i>Learning Rate</i>	1e-4

Como uma forma de acompanhar a curva de convergência do treinamento e visualizar como ela se comporta, foi utilizada uma ferramenta chamada *Tensorboard*. Foi obtida a curva de erro de treinamento (Figura 23). Na curva de treino, apesar da instabilidade, é possível verificar que na época 60 o modelo atingiu a convergência, em direção ao erro mínimo. O número de épocas foi escolhido com base em alguns trabalhos: (AKYAZI; EBRAHIMI, 2019a, 2019b; ORORBIA et al., 2019; MISHRA; SINGH; SINGH, 2020; LIU et al., 2021; YASUKAWA et al., 2021; QIAN et al., 2022; DIAO; DING; TAROKH, 2020).

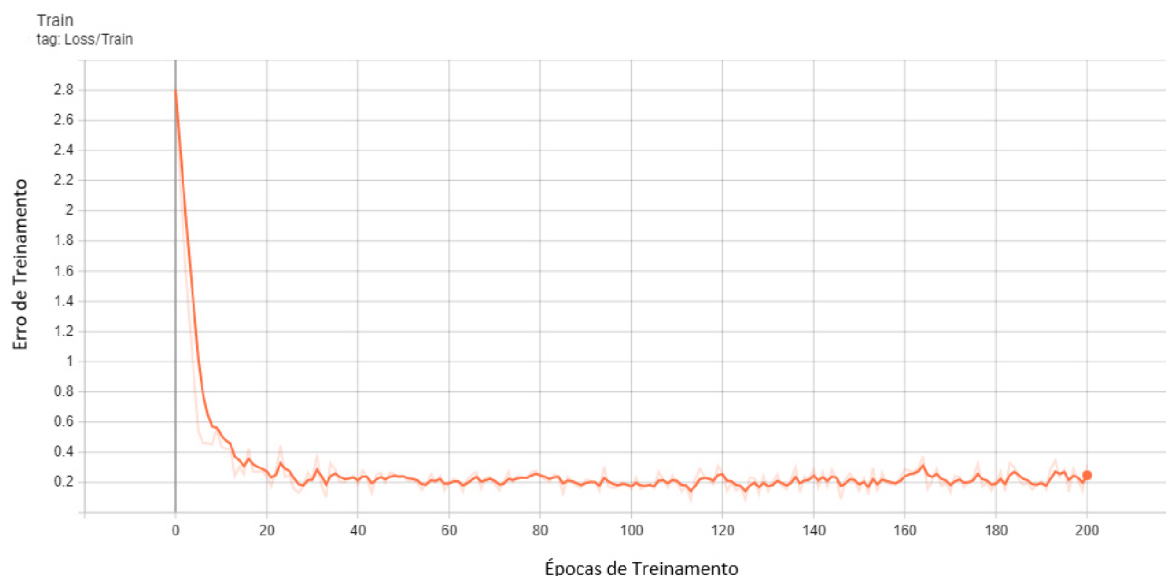


Figura 23: Curva de Convergência de Treino (*Autoencoder Variacional com Hyperprior* Treinado com Imagens Subaquáticas)

A ferramenta *Tensorboard* foi utilizada para acompanhar a reconstrução da imagem de saída do modelo a cada época. A Figura 24 mostra uma comparação entre imagem original, imagem de saída do modelo após época 1 e imagem de saída do modelo após época 200. É possível perceber que após a época 1 a saída apresentou uma imagem com um certo embaçamento. Contudo, após a época 200 a imagem já estava mais semelhante a original. A Figura 25 mostra um zoom nas imagens apresentadas na Figura 24.

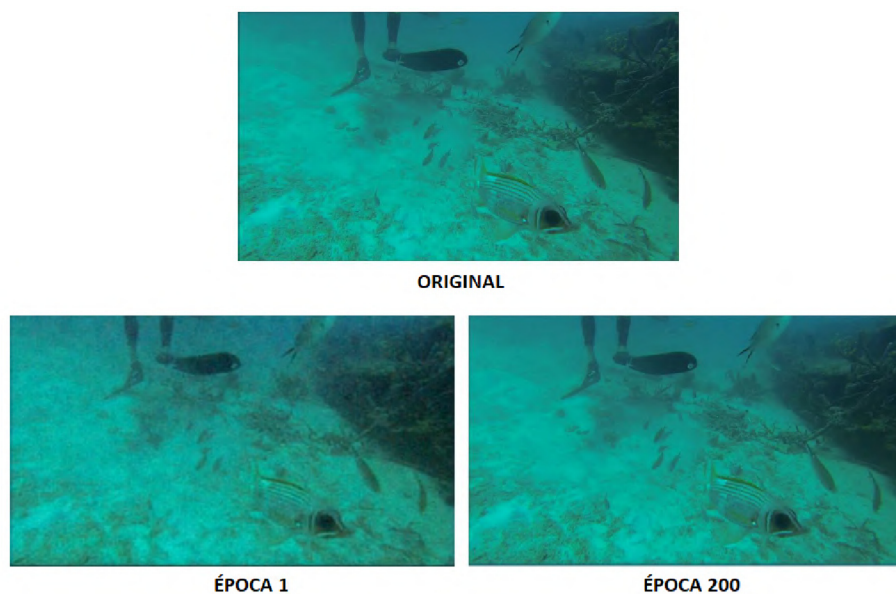


Figura 24: Acompanhamento da Saída do Modelo - *Autoencoder Variacional com Hyperprior* Treinado com Imagens Subaquáticas

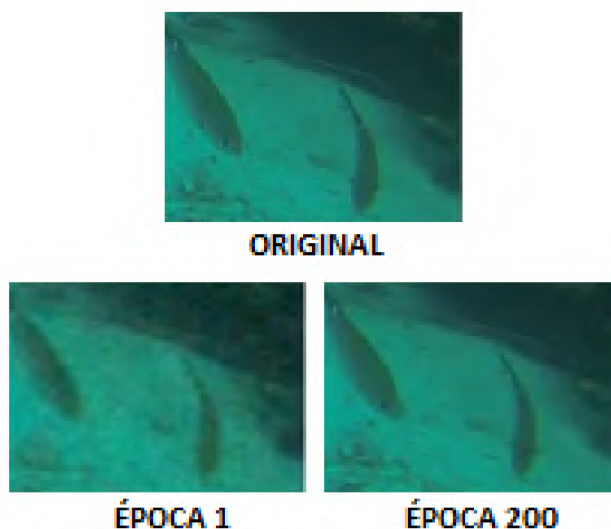


Figura 25: Acompanhamento da Saída do Modelo - *Autoencoder* Variacional com *Hyperprior* Treinado com Imagens Subaquáticas - Aproximado

Um ponto importante a se analisar é o espaço latente resultante do modelo, visto que está diretamente relacionado com a taxa de compressão. Como apresentado, a entropia está associada ao quão rara é uma informação dentro de um sinal. As Figuras 26(a) e 26(b) mostram, respectivamente, o espaço latente na época 1 e na época 200. É possível observar que o espaço latente da época 200 possui baixa entropia, ou seja, a variação de informação entre os canais é pouca, o que acarreta em muita repetição, tornando a informação mais compressível, otimizando o processo de compressão (informação que repete muito é comprimida em menos *bits*). Enquanto que, o espaço latente da época 1 possui alta entropia, pois é possível ver muita variação de verde para vermelho, o que torna a informação pouco compressível devido a essa variabilidade (alta quantidade e variedade de informação)⁵. Então, se uma imagem é toda branca, ela possui baixa entropia, pois a diversidade de informação presente é pouca e toda igual. Entretanto, se uma imagem é cheia de cores, ela possui alta entropia, pois possui muita diversidade de informação, sendo assim, difícil de codificar com poucos *bits*. Ainda, ao analisar o canal com maior entropia na época 200 (canal mais amarelo - primeira coluna), percebe-se a existência de mais informação, que possivelmente é pouco relacionada a água e mais relacionada as estruturas presentes na imagem (imagem da Figura 24).

⁵Nessa codificação: Azul = 0 (pouca informação) e Vermelho = 1 (muita informação)

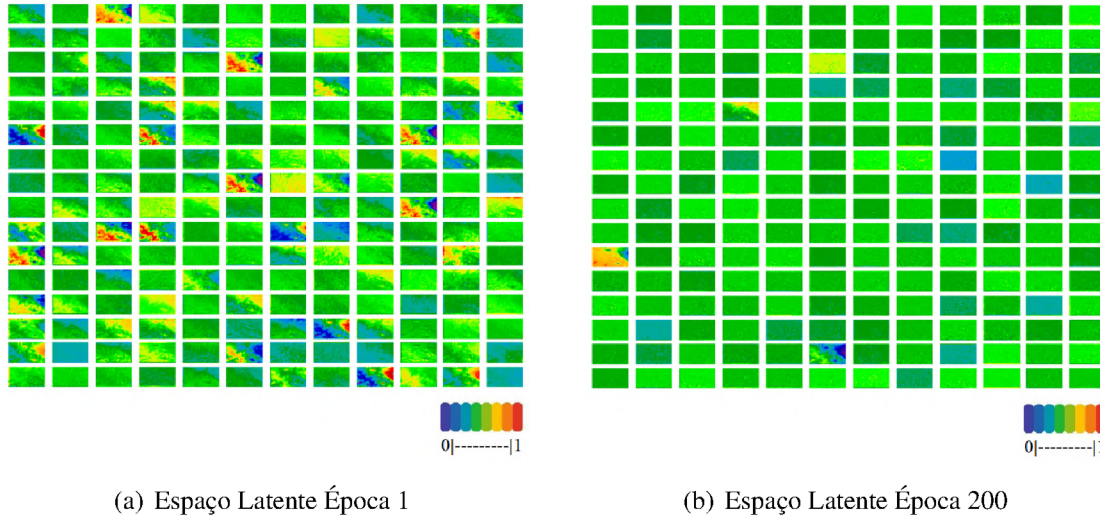


Figura 26: *Autoencoder Variacional com Hyperprior* Treinado com Imagens Subaquáticas

Percebe-se nas Figuras 26(a) e 26(b) a notória diferença entre os espaços latentes. Na época 200 as imagens são mais "brandas" apresentando vários canais com baixa entropia (verde e azul). Somente alguns canais codificam toda a informação em alta entropia, por exemplo: canal mais amarelo - primeira coluna. A ideia do método de compressão é codificar a informação de maneira que a entropia entre os canais do espaço latente seja baixa, ou seja, a maioria dos canais estão nessa tonalidade de verde e azul, o que representa pouca quantidade de informação.

As imagens de entrada da rede estão na forma $(N, 3, H, W)$, ou seja, $(16, 3, 1080, 1920)$. Usualmente, redes para compressão de imagens por padrão dividem a resolução da imagem por 16, resultando na seguinte representação no espaço latente: $(N, 192, 1080/16, 1920/16)$, onde 192 é o número de canais, como já explicado na Subseção 6.1. Desta forma, é possível calcular o CR (*Compression Ratio*) resultante, dado em (1):

Assim,

$$CompressionRatio = \frac{1920 \times 1080}{\frac{1920}{16} \times \frac{1080}{16}} = 256 \quad (10)$$

A imagem no espaço latente foi comprimida 256 vezes. Os valores de PSNR e BPP obtidos, após avaliação com o *dataset* de validação, podem ser visualizados na Tabela 5.

6.3.3 Ajuste Fino na Rede Original Pré-Treinada (Fine Tuning)

Um ajuste fino com base no *dataset subaquático* (subseção 6.2) foi também aplicado à rede original *autoencoder* variacional com *hyperprior* (treinada com o *dataset* do teste 6.3.1). Os parâmetros utilizados foram os mesmos da Tabela 4.

A Figura 27 mostra uma comparação entre imagem original e imagem de saída do modelo após época 200, com o Ajuste Fino na Rede Original Pré-Treinada. Percebe-se que a imagem de saída, na época 200, está bem semelhante a imagem original. A Figura 28 mostra uma parte aproximada de cada imagem apresentada na Figura 27.



Figura 27: Acompanhamento da Saída do Modelo - Ajuste Fino na Rede Original Pré-Treinada

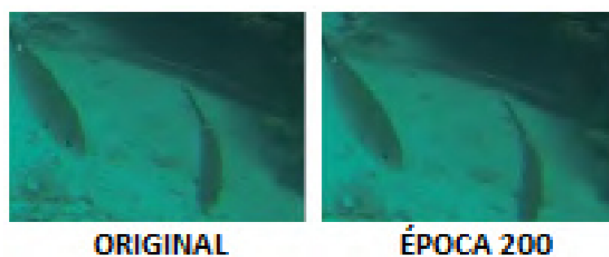


Figura 28: Acompanhamento da Saída do Modelo - Ajuste Fino na Rede Original Pré-Treinada - Aproximado

As Figuras 29(a) e 29(b) mostram, respectivamente, os espaços latentes na época 1 e na época 200 com o Ajuste Fino na Rede Original Pré-Treinada. A Figura 29(b) apresenta alguns canais vermelhos e amarelos, ou seja, são canais onde maior quantidade e diversidade de informação da imagem está armazenada.

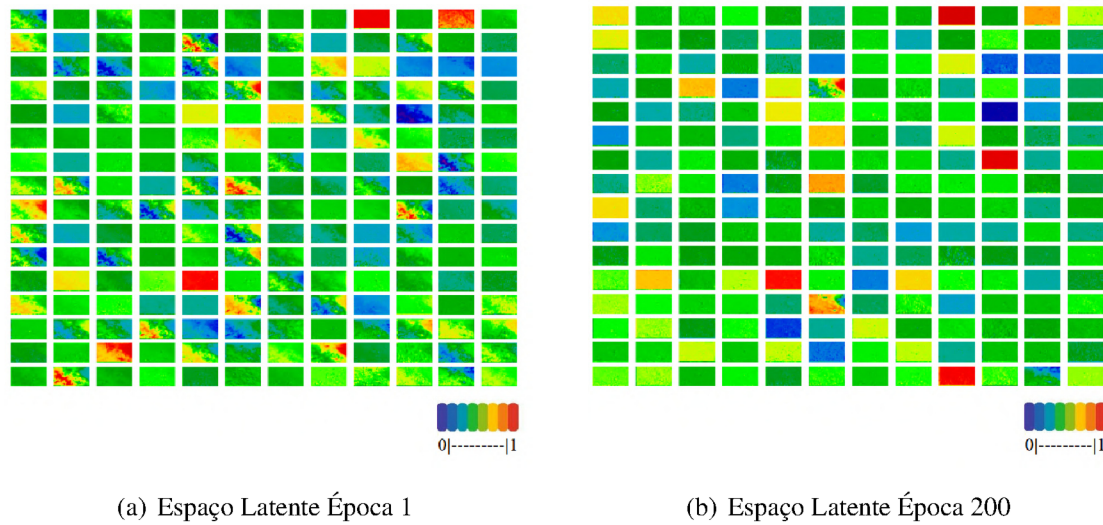


Figura 29: Ajuste Fino na Rede Original Pré-Treinada

Percebe-se uma diferença entre o espaço latente (Figura 26(b)) após o término de treinamento com o experimento apresentado na Subseção 6.3.2, e o espaço latente (Figura 29(b)) após o término de treinamento com o Ajuste Fino na Rede Original Pré-Treinada. Enquanto que, o primeiro foi treinado totalmente com imagens subaquáticas somente, o segundo foi pré-treinado com imagens não subaquáticas e com o novo treino, os pesos da rede foram ajustados com base nas novas imagens adicionadas ao treinamento, ou seja, com as imagens subaquáticas.

As imagens não subaquáticas, que possuem muita diversidade de informação (comparadas com imagens subaquáticas), quando comprimidas, necessitam de mais canais para guardar as características que as compõem para representar a diversidade de informação. Então, a rede pré-treinada com imagens não subaquáticas e retreinada com imagens subaquáticas pode ter aprendido muito da experiência anterior e replicado o entendimento do aprendizado no novo treino. Por essa razão, percebem-se mais canais vermelhos e amarelos na Figura 29(b) do que na Figura 26(b). Como na subseção anterior, a imagem no espaço latente foi comprimida 256 vezes.

Os valores de PSNR e BPP obtidos, após avaliação com o *dataset* de validação, podem ser visualizados na Tabela 5. Vale a pena destacar que foram ainda adicionados somente a tabela, a avaliação referentes aos *codecs*: JPEG e WebP.

Categoria	Original		Totalmente Treinado		Fine Tuning		JPEG		WebP	
	PSNR	BPP	PSNR	BPP	PSNR	BPP	PSNR	BPP	PSNR	BPP
<i>Só Água</i>	40,129	0,162	36,393	0,138	37,802	0,136	44,563	0,501	41,562	0,260
<i>Meio Termo</i>	37,111	0,205	33,870	0,168	35,236	0,171	40,446	0,690	38,784	0,361
<i>Praticamente Sem Água</i>	32,293	0,488	27,667	0,308	29,336	0,405	35,148	1,426	34,464	1,041
<i>Indoor</i>	35,176	0,290	30,016	0,253	32,893	0,277	39,011	0,974	37,363	0,567
<i>Outdoor</i>	30,923	0,668	25,955	0,327	28,465	0,526	35,837	1,652	34,413	1,355

Tabela 5: Valores de PSNR e BPP dos Experimentos Apresentados

Para melhorar a visualização dos resultados de PSNR e BPP obtidos nos testes realizados, foi plotado um gráfico, conforme a Figura 30. Quanto maior for o valor do PSNR maior é a qualidade visual da imagem, ou seja, a diferença entre imagem de entrada e imagem de saída é praticamente imperceptível. A partir de um PSNR = 30, já não se nota diferença entre as imagens. Já o BPP, quanto menor for o valor, mais compressão foi aplicada a imagem.

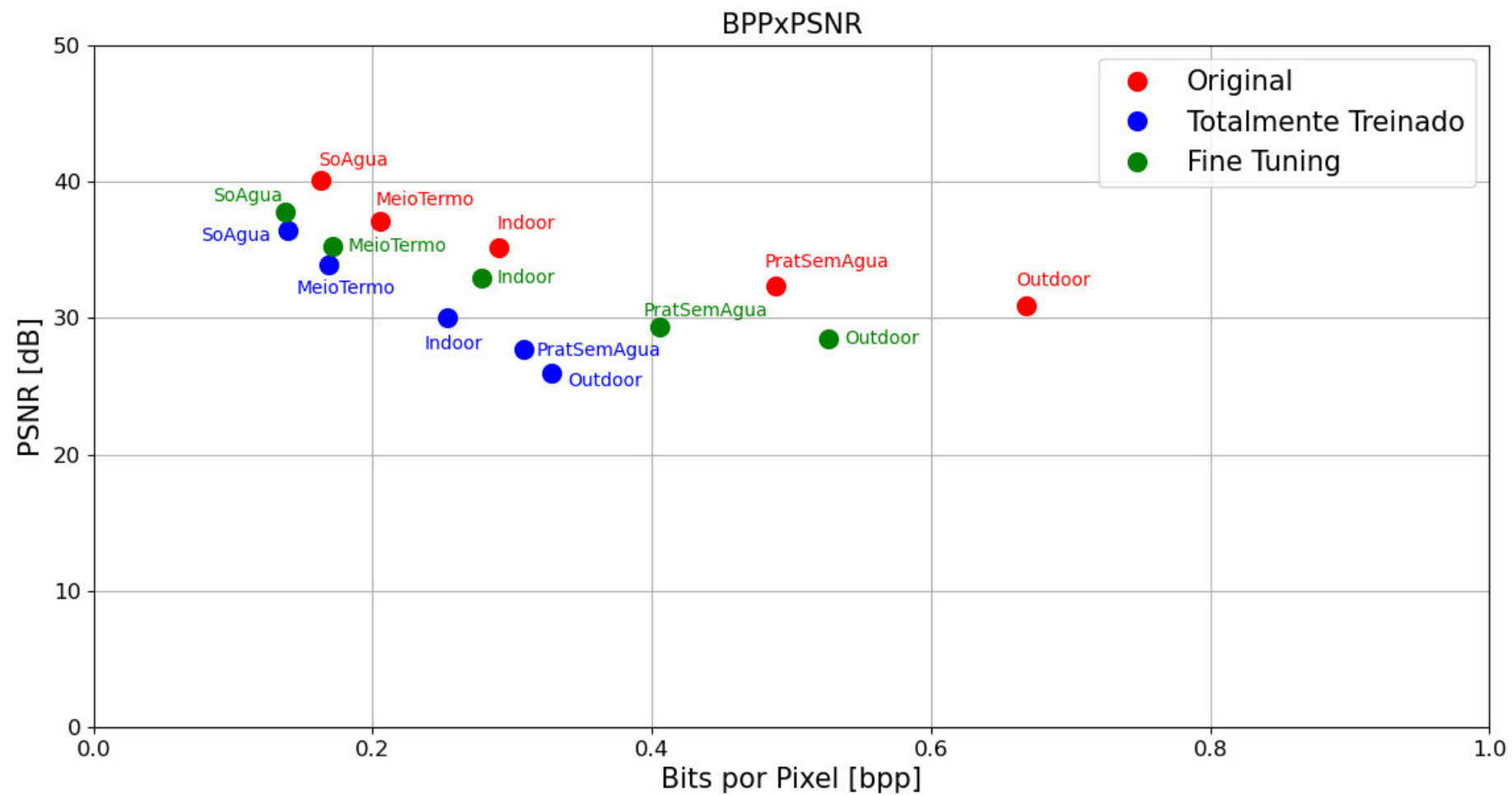


Figura 30: Gráfico BPPxPSNR

Ao analisar tanto a Tabela 5, quanto a Figura 30 algumas conclusões são obtidas. Percebe-se que os *codecs* baseados em aprendizado superam, em questão de compressão, os *codecs* tradicionais: JPEG e WebP. Uma análise mais geral mostra que o experimento que teve mais ganho na compressão, quando comparado com o experimento original apresentado na subseção 6.3.1, foi o experimento que foi treinado com imagens subaquáticas, na subseção 6.3.2, pois comprimiu mais e manteve uma boa qualidade visual. Com um PSNR acima de 30 (quanto maior melhor) é difícil perceber diferença entre a imagem de entrada e a imagem reconstruída.

Os *datasets* *Só Água* e *Meio Termo* são melhor tratados pelos experimentos Ajuste Fino na Rede Original Pré-Treinada e *Autoencoder* Variacional com *Hyperprior* Treinado com Imagens Subaquáticas, sendo que a perda de qualidade apresentada no último experimento é existente, mas não prejudicial. Essa perda de qualidade não é tão relevante em muitas aplicações, visto que comprimir mais imagens que tem só água justifica-se por não haver muita informação neste tipo de imagem. A perda de qualidade mais presente nos resultados do experimento *Autoencoder* Variacional com *Hyperprior* Treinado com Imagens Subaquáticas talvez se deva ao fato de que, o mesmo é treinado totalmente com menos de 10.000 imagens, uma quantidade de imagens que não é a apropriada para esse tipo de rede.

Ao observar a taxa de compressão dos dois primeiros métodos descritos nas subseções 6.3.1 e 6.3.2, na categoria *Outdoor*, há uma boa diminuição da taxa de compressão (menos da metade do valor), de 0,668 para 0,327, até por isso que houve perda de qualidade. Já na categoria *Indoor* existe pouca diferença entre as taxas de compressão dos métodos, de aproximadamente 0,04. Analisando a categoria *Praticamente Sem Água*, onde tem mais informação, percebe-se um ganho maior de compressão (de 0,488 para 0,308). Então, pode-se concluir que em uma operação embaixo d'água, onde não se terá somente monotonia da água, é possível alcançar taxas maiores de compressão.

Comparando os pontos vermelho, azul e verde da categoria *Só Água* versus os pontos vermelho, azul e verde da categoria *Outdoor*, percebe-se que as diferenças entre as duas categorias são muito maiores na compressão, visto que há mais impacto na compressão de imagens *Outdoor* do que em subaquáticas. Já nos pontos correspondentes ao *Indoor* há praticamente somente perda de qualidade e há pouca compressão extra. Por fim, outra consideração é que o ganho de compressão, nos dois primeiros experimentos, na categoria *Só Água* não foi muito, ele foi de 0,162 para 0,138.

6.3.4 Avaliação da Variação do Lambda no Desempenho do CODEC

Neste experimento busca-se avaliar o desempenho da rede frente a modificações do valor do λ . Busca-se propor um método onde seja possível ajustar o valor de λ em função da quantidade de informação presente na imagem a ser comprimida. A ideia é que quando a imagem é mais monótona, a exemplo as imagens da categoria *Só Água*, é possível comprimi-las mais, já que a degradação provocada pouco afeta a interpretação da imagem. A Tabela 5 mostra que para a categoria *Só Água*, no experimento da subseção 6.3.2, é possível obter um PSNR de 36,393 e uma taxa de compressão de 0,138. Como dito, com um PSNR a partir de 30, já não se nota muita diferença de qualidade visual entre a imagem de entrada e a imagem reconstruída. Neste contexto, é possível reduzir o valor do PSNR, que por consequência reduzirá o valor da taxa de *bits*. Quando se está fazendo uma operação com robô subaquático, uma grande porcentagem das imagens que o equipamento adquire são da classe *Só Água*. Essa informação poderia estar sendo comprimida, então nesta subseção busca-se avaliar o impacto de diferentes valores de λ escolhidos manualmente no resultado da rede.

Para fazer a análise do impacto de diferentes λ s na taxa de compressão e na qualidade visual da imagem, três experimentos envolvendo treinamento e avaliação da rede foram especificados. Foram realizados novos treinos de ajuste fino na rede original pré-treinada, com adição de 630 imagens subaquáticas alterando os valores λ s. Foram realizados três treinamentos diferentes, onde cada treinamento utilizou um *dataset* com uma única categoria de imagem (*Só Água*, *Meio Termo* e *Praticamente Sem Água*) e λ s diferentes.

Assim, para os três experimentos optou-se atribuir arbitrariamente (tendo como base o que foi explicado na seção 4.2 a respeito do ajuste de compensação taxa-distorção): $\lambda = 1/20$ (para categoria *Só Água*), $\lambda = 1/5$ (para categoria *Meio Termo*) e $\lambda = 1$ (para categoria *Prat Sem Água*). Para exemplificar, ao atribuir $\lambda = 1/20$ diminui-se a importância da distorção, ou seja, a imagem pode ficar mais distorcida, já que ela perde a qualidade visual e comprime mais. Neste caso, a Tabela 6 mostra a especificação de cada treinamento realizado. Vale a pena ressaltar que, as imagens obtidas para cada grupo, também foram retiradas de alguns vídeos e de um *dataset* específico de imagens subaquáticas⁶⁷⁸⁹¹⁰.

⁶Vídeo 4 - Canal: Biodun Onipede (Link: https://www.youtube.com/watch?v=HoYOB18LMS4ab_channel=BiodunOnipede)

⁷Vídeo 5 - Canal: Nautilus LIB.. (Link: <https://www.youtube.com/watch?v=gpeTipmcwIQ>)

⁸Vídeo 6 - Canal: Hansueli Holzer (Link: <https://www.youtube.com/watch?v=hXLIA0lafks>)

⁹Vídeo 7 - Canal: Nafees - Nature Videos 8K (Link: <https://www.youtube.com/watch?v=Qr1ueT5Jnqk>)

¹⁰Dataset Específico: https://li-chongyi.github.io/proj_benchmark.html

Tabela 6: Três Treinos com Grupos de Imagens e *Lambdas* Distintos

	Parâmetro	Valor
GRUPO 1: $\Lambda = 1/20$ para imagens <u>Só Água</u>	Número de Épocas	200
	Imagens de Treino (80%)	504
	Imagens de Teste (20%)	126
	<i>Batch Size</i>	16
	<i>Learning Rate</i>	$1e-4$
	<i>Lambda</i>	$1/20$

	Parâmetro	Valor
GRUPO 2: $\Lambda = 1/5$ para imagens <u>Meio Termo</u>	Número de Épocas	200
	Imagens de Treino (80%)	504
	Imagens de Teste (20%)	126
	<i>Batch Size</i>	16
	<i>Learning Rate</i>	$1e-4$
	<i>Lambda</i>	$1/5$

	Parâmetro	Valor
GRUPO 3: $\Lambda = 1$ para imagens <u>Prat Sem Água</u>	Número de Épocas	200
	Imagens de Treino (80%)	504
	Imagens de Teste (20%)	126
	<i>Batch Size</i>	16
	<i>Learning Rate</i>	$1e-4$
	<i>Lambda</i>	1

As Figuras apresentadas a seguir tem objetivo de demonstrar como a imagem de saída se comporta, frente a diferentes valores de λ . As Figuras 31 e 32 correspondem a saída do modelo, com um λ de $1/20$, ou seja, a distorção (D) está sendo minimizada. A Figura 31 corresponde a categoria *Só Água*, onde percebe-se que mesmo com alta taxa de compressão, perceptualmente se percebe pouca perda de qualidade, pois a diversidade de informação presente na imagem é pouca. Então, mesmo com altas taxas de compressão, a aparente perda de qualidade é baixa. Já a Figura 32 corresponde a categoria *Meio Termo* (somente para exemplificação), onde percebe-se perceptualmente que houve uma perda de qualidade maior, pois a imagem tem mais diversidade de informação. Então, para o caso de imagens meio termo, ao aplicar uma alta taxa de compressão, é percebida uma maior perda de qualidade na imagem.

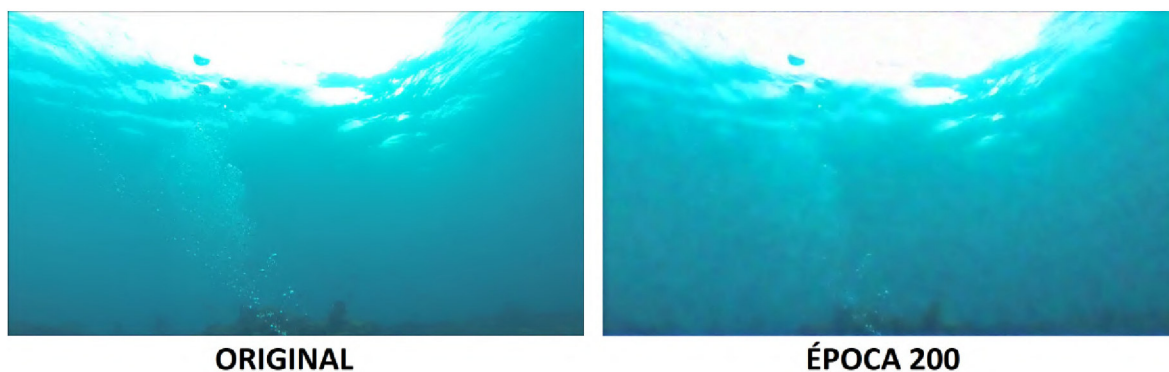


Figura 31: Imagem da Categoria *Só Água* com λ $1/20$



Figura 32: Imagem da Categoria *Meio Termo* com λ $1/20$

A Figura 33 corresponde a saída do modelo, com um λ de $1/5$. A imagem faz parte da categoria *Meio Termo*, onde percebe-se que a perda de qualidade foi menor do que na Figura 32. Isso se deve ao fato de que, o peso aplicado para minimizar a distorção foi um valor maior, ou seja, está se levando mais em conta a diferença entre a imagem de entrada e a imagem de saída, resultando em uma imagem de melhor qualidade visual.



Figura 33: Imagem da Categoria *Meio Termo* com λ $1/5$

Por fim, a Figura 34 corresponde a saída do modelo, com um λ de 1, ou seja, a perda de qualidade visual da imagem é imperceptível, o que é constatado através da imagem da categoria *Praticamente Sem Água* apresentada.

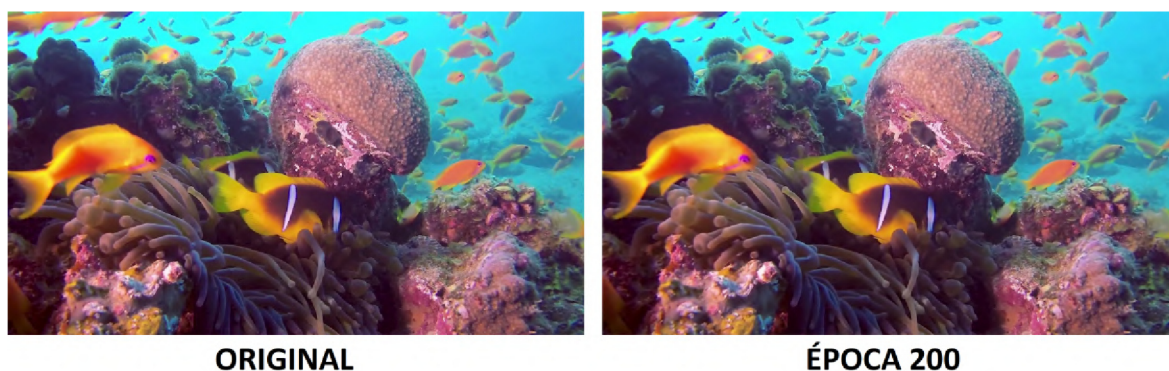


Figura 34: Imagem da Categoria *Praticamente Sem Água* com λ 1

Os valores de PSNR e BPP obtidos, para os três experimentos apresentados, após avaliação com o *dataset* de validação, podem ser visualizados na Tabela 7.

Categoria	Método: Lambda 1/20		Método: Lambda 1/5		Método: Lambda 1	
	PSNR	BPP	PSNR	BPP	PSNR	BPP
<i>Só Água</i>	30,285	0,062	35,256	0,084	39,997	0,167
<i>Meio Termo</i>	29,004	0,069	33,196	0,094	37,168	0,210
<i>Praticamente Sem Água</i>	23,833	0,152	28,436	0,227	32,404	0,501
<i>Indoor</i>	24,691	0,114	30,862	0,155	34,924	0,317
<i>Outdoor</i>	22,672	0,215	27,010	0,297	30,989	0,694

Tabela 7: Valores de PSNR e BPP dos Três Treinamentos com Diferentes *Lambdas*

Para melhor visualização dos resultados de PSNR e BPP obtidos, foi plotado um gráfico, conforme a Figura 35.

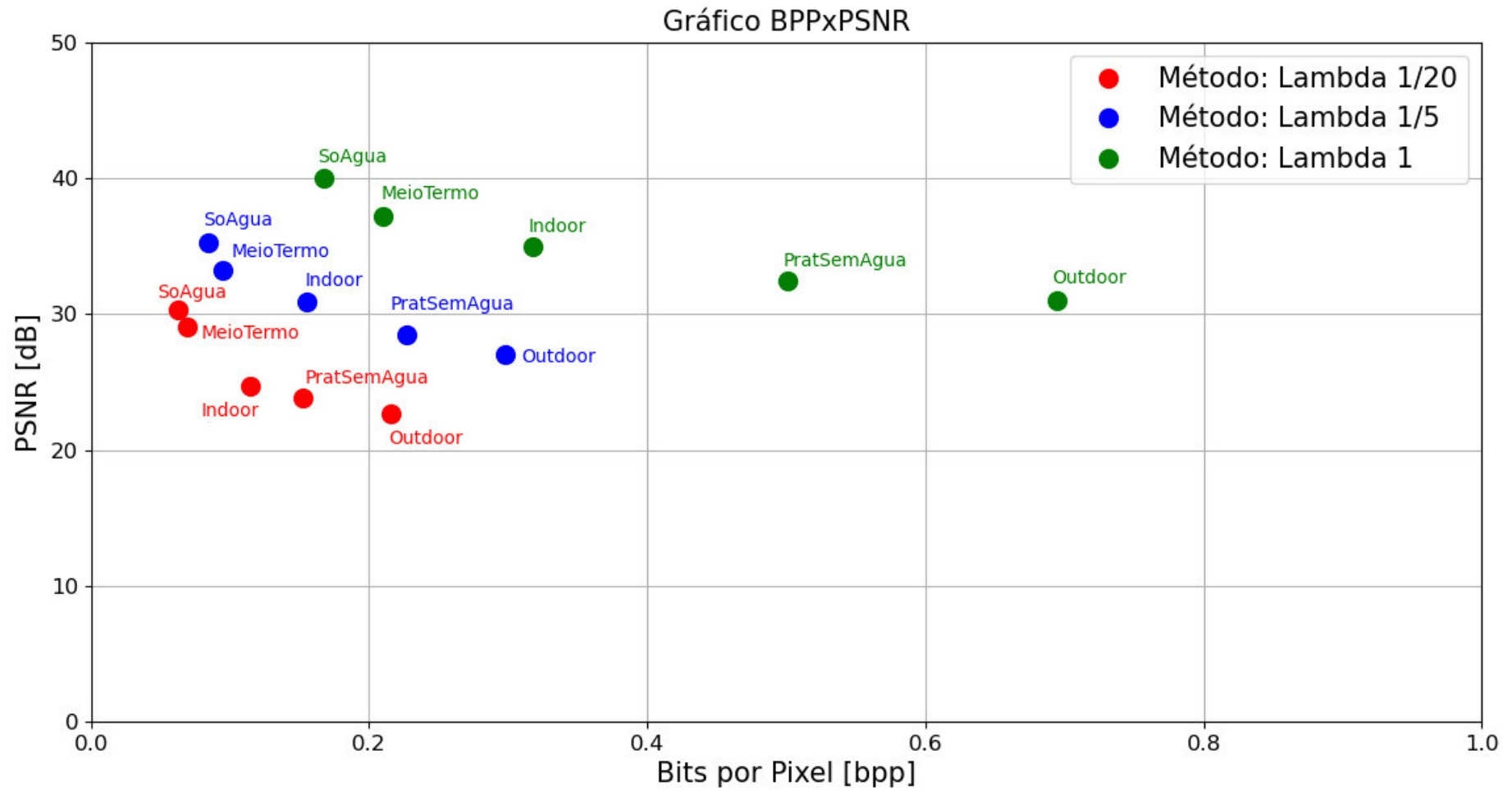


Figura 35: Gráfico BPPxPSNR - *Lambdas* Diferentes

O parâmetro *Lambda* pode ser visto como um parâmetro de qualidade, associado ao impacto da distorção no processo de compactação. Analisando a Tabela 7, quanto mais a direita, mais se valoriza a qualidade, ou seja, obtém-se um valor maior de PSNR. Contudo, quanto mais a esquerda, mais se valoriza a taxa de compressão, ou seja, os valores de BPP são menores. Observando tanto a Tabela 7 quanto o Gráfico na Figura 35, um PSNR de 30,285 na categoria *Só Água* é adequado, visto que a maioria das imagens subaquáticas, em missões com AUVs por exemplo, usualmente são *Só Água*. Se elas são *Só Água*, é possível trabalhar com um *lambda* na primeira faixa da esquerda, pois ele mantém uma boa qualidade visual e um baixo valor de taxa de compressão, sendo ele de 0,062. No caso do grupo Método: $\lambda = 1/20$, observando a categoria *Só Água*, a imagem pode perder a qualidade visual, já que, como comentado, esse tipo de imagem não apresenta muita informação. Porém, ao avaliar a categoria *Praticamente Sem Água* o valor de PSNR é de aproximadamente 24, o que mostra que para essa categoria a perda da qualidade visual atrapalha, ou seja, a imagem de saída não é tão semelhante a imagem de entrada.

Os resultados obtidos respaldam a hipótese de que a rede possa aprender que quanto menos informação na imagem, maior a possibilidade de compressão. Então, desenvolver e avaliar um código que faça isso de maneira automática, pode ser pertinente. Manter um PSNR com um valor que não prejudique a qualidade visual, para todos os casos, pode trazer grande taxa de compressão principalmente quando se trabalha com imagens com características de monotonia, como em imagens *Só Água*.

Dito isso, na próxima subseção são apresentados os testes e análises utilizando a proposta deste trabalho, onde ajusta-se a compensação taxa-distorção a partir da aplicação do descritor SIFT à imagem.

6.3.5 Ajuste Fino na Rede Original Pré-Treinada Sem e Com Inclusão do SIFT

Como citado no início da seção 6.3 o *dataset* de treinamento, utilizado para o presente teste, sofreu algumas alterações e sua distribuição em relação a medida do SIFT foi mostrada na Figura 20(b). Nesse contexto, optou-se por além de fazer o treinamento da rede com a inclusão do SIFT na *Loss*, também refazer o teste da subseção 6.3.3 (sem inclusão do SIFT), com o novo conjunto de dados para justa comparação.

Dois testes foram realizados: o primeiro *Fine Tuning Sem SIFT*, onde o teste da subseção 6.3.3 foi repetido com o novo *dataset* de treino e o segundo *Fine Tuning Com SIFT*, onde a equação (9) foi utilizada como função de perda. A Tabela 8 apresenta os parâmetros utilizados para o treinamento.

Tabela 8: Parâmetros de Treinamento - Ajuste Fino na Rede Original Pré-Treinada Sem e Com Inclusão do SIFT

Parâmetro	Valor
Número de Épocas	200
Imagens de Treino (80%)	7756
Imagens de Teste (20%)	1939
<i>Batch Size</i>	16
<i>Learning Rate</i>	1e-4

Os valores de PSNR e BPP obtidos, após avaliação com o *dataset* de validação, podem ser visualizados na Tabela 9.

Categoria	Original		Fine Tuning Sem SIFT		Fine Tuning Com SIFT	
	PSNR	BPP	PSNR	BPP	PSNR	BPP
<i>Só Água</i>	41,795	0,124	39,453	0,084	38,623	0,072
<i>Meio Termo</i>	37,111	0,205	35,403	0,162	34,716	0,175
<i>Praticamente Sem Água</i>	32,293	0,488	29,897	0,397	30,047	0,408
<i>Indoor</i>	35,176	0,290	32,715	0,275	32,961	0,277
<i>Outdoor</i>	30,923	0,668	28,348	0,513	28,628	0,534

Tabela 9: Valores de PSNR e BPP do Treino Sem e Com Inclusão do SIFT

Para melhor visualização dos resultados de PSNR e BPP obtidos, foi plotado um gráfico, conforme a Figura 36.

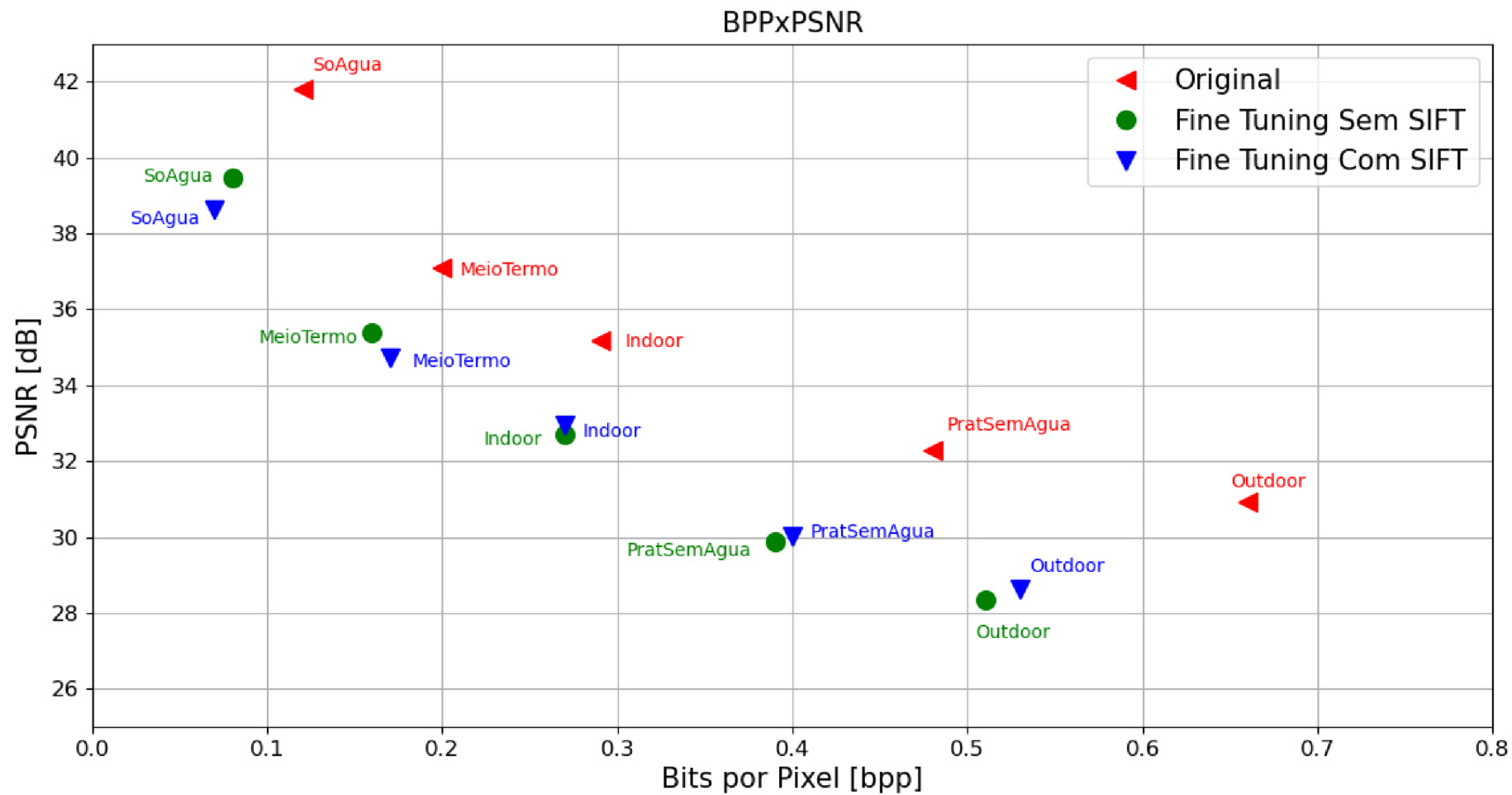


Figura 36: Gráfico BPPxPSNR - Sem e Com Inclusão do SIFT

Ao analisar a Tabela 9 e a Figura 36, pode-se observar que tanto o teste *Fine Tuning Sem SIFT* quanto o teste *Fine Tuning Com SIFT* obtiveram maior taxa de compressão, quando comparado ao *benchmark* estabelecido, em todas as categorias. O SIFT foi utilizado como ferramenta para analisar e quantificar a variabilidade das informações presentes na imagem. Ao comparar o *Fine Tuning Sem SIFT* com o *Fine Tuning Com SIFT* percebe-se que em termos de BPP há pouca diferença entre eles. No entanto, é possível notar que há um pequeno ganho na taxa de compressão no experimento *Fine Tuning Com SIFT* na categoria *Só Água*, ou seja, aumentou a compressão deste tipo de imagem.

A Figura 37 mostra a relação entre o SIFT, PSNR e BPP (valores do experimento: *Fine Tuning Com SIFT* da Tabela 9). É possível perceber que quanto menor o valor de SIFT (categoria *Só Água*), mais compressão é alcançada e a qualidade visual é adequada. Já as categorias *Praticamente Sem Água*, *Indoor* e *Outdoor*, que possuem uma média de valor de SIFT semelhante e são imagens com maior quantidade de informação, não alcançam grande compressão, quando comparado com a categoria mais monótona, como também perdem um pouco de qualidade visual, mas nada que prejudique o entendimento da cena.

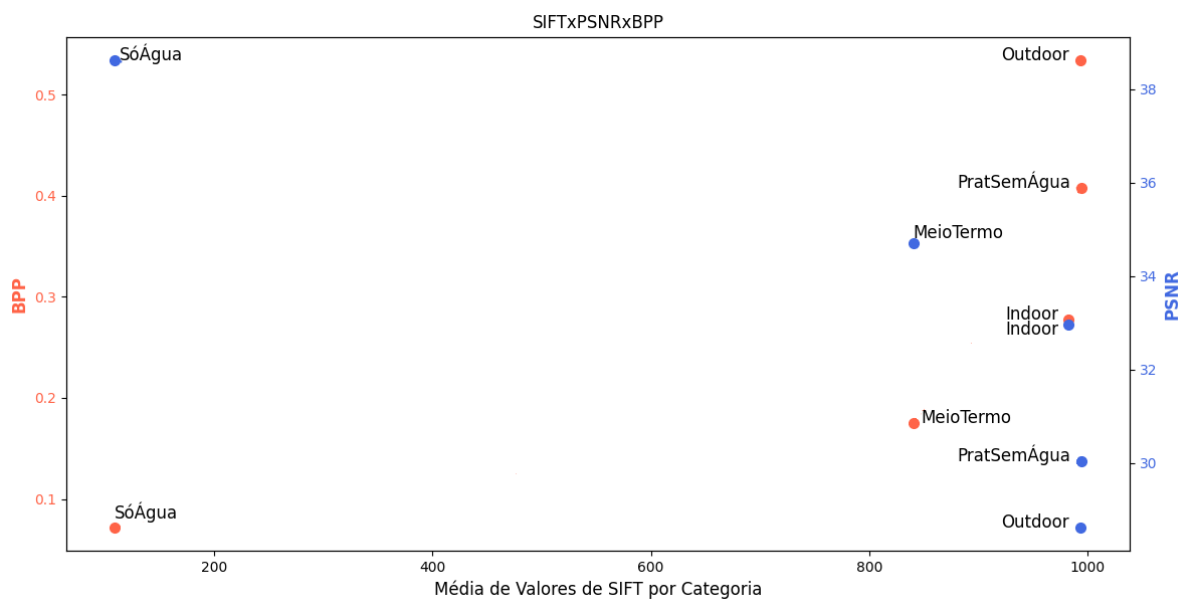


Figura 37: SIFTxPSNRxBPP

Como outra forma de avaliação, foram selecionados dois vídeos com 25 segundos cada e a sequência de *frames* resultante foi avaliada. O primeiro vídeo é composto apenas por imagens dentro da categoria *Só Água* ($V_{SóÁgua}$). Enquanto o segundo vídeo reflete imagens subaquáticas dentro das categorias *Meio Termo* e *Praticamente Sem Água* (V_{Geral}). A Tabela 10 mostra os valores obtidos de PSNR e BPP, bem como o Tamanho em *Bytes* (TB) estimado da imagem de saída, que é definido pela quantidade de *pixels* e a profundidade de cor de cada *pixel*. Ambos os vídeos originalmente (sem compressão) ocupam aproximadamente 4460 MB, enquanto que no *Fine Tuning Com SIFT*, respectivamente, ambos passaram a ocupar 13 e 51 MB.

Vídeo	Original			Fine Tuning Sem SIFT			Fine Tuning Com SIFT		
	PSNR	BPP	TB (MB)	PSNR	BPP	TB (MB)	PSNR	BPP	TB (MB)
$V_{SóÁgua}$	43,132	0,114	21	41,337	0,076	14	41,137	0,071	13
V_{Geral}	34,997	0,322	59	34,261	0,287	53	34,113	0,279	51

Tabela 10: Análise dos *Frames* de um Vídeo

7 CONSIDERAÇÕES FINAIS

O objetivo geral deste trabalho foi estudar e tratar o problema da codificação da transmissão de imagem subaquática, visando principalmente métodos que trabalham com Redes *Autoencoder* (*encoder-decoder*). Constatou-se a importância do assunto no universo da indústria de petróleo e gás, por exemplo, a existência de trabalhos relacionados a temática, como também de ferramentas para o desenvolvimento da proposta.

Foi realizada uma revisão dos principais conceitos e ferramentas utilizadas para o problema de *codec*. Um levantamento do estado da arte deste tema também foi realizado e conduziu a escolha de uma rede *encoder-decoder* como metodologia para tratamento do problema. Foi proposta uma alternativa para obtenção de imagens subaquáticas para a construção de um *dataset* de imagens subaquáticas.

O comportamento da rede *Autoencoder* Variacional com *Hyperprior* proposta foi analisado do ponto de vista de desempenho em termos de redução dimensional para transmissão, como também qualidade da imagem de saída e nível de compressão atingido. A sequência de experimentações realizadas permitiram a apropriação e entendimento dos conceitos.

Portanto, uma proposta de rede da literatura, para compressão de imagens, foi recuperada e seu possível impacto foi analisado. Após essa análise, parâmetros da rede foram alterados com o objetivo de se obter qualidade visual e taxa de compressão adequados a aplicação. Através dos experimentos de avaliação da variação do λ no desempenho do *codec* percebeu-se que este parâmetro pode ser melhor ajustado em função da quantidade de informação presente na imagem. Entendido isso, foi proposto um método baseado, também, na Rede *Autoencoder* Variacional, mas em conjunto com uma métrica capaz de avaliar a quantidade de informação presente em uma imagem.

Como proposta, utilizou-se o SIFT por este ser utilizado para quantificar a quantidade de informação na imagem, sendo possível definir o quanto comprimir a imagem, mantendo-a com boa qualidade visual. O SIFT não é uma métrica específica para quantificar informações em imagens, mas os primeiros resultados mostraram que existe a possibilidade de dar continuidade a este estudo, buscando ou até mesmo desenvolvendo outras métricas para este propósito.

Assim, ao utilizar uma arquitetura de compressão, treinada com imagens subaquáticas, é possível obter baixas taxas de *bits* e boa qualidade visual, principalmente em imagens dentro das categorias *Só Água* ou *Meio Termo*. Em relação ao tipo de imagem onde há predominância de água (*Só Água*), todos os experimentos mostraram que mesmo com alta compressão (baixo bpp), a qualidade de reconstrução da imagem de saída não foi perdida, validando a proposta como uma técnica para quantificar a quantidade geral de informação em imagens subaquáticas a serem compactadas.

O presente trabalho foi realizado com apoio financeiro do Programa de Recursos Humanos da Agência Nacional do Petróleo, Gás Natural e Biocombustíveis – PRH-ANP, suportado com recursos provenientes do investimento de empresas petrolíferas qualificadas na Cláusula de P, D&I da Resolução ANP nº 50/2015. Agradeço ao PRH22 e seus coordenadores, os professores: Eder Mateus Nunes Gonçalves, Maria Isabel Corrêa da Silva Machado e Silvia Silva da Costa Botelho por todo suporte sempre quando necessário.

7.1 Trabalhos Futuros

Para trabalhos futuros enumera-se algumas sugestões. Os tópicos a seguir as elencam:

- Executar uma tarefa de pré-processamento nas imagens para melhoria (ex: tratar a turbidez como pré-processamento). Por vezes, quando há melhora da imagem antes da compressão, essa ação facilita a tarefa do compressor;
- Melhorar a distribuição e balanceamento das amostras por categorias de imagens no *dataset* subaquático, refletindo as características de imagens que são capturadas durante uma missão com veículo subaquático não tripulado;
- Utilizar outros métodos quantitativos que indiquem a relação entre imagem original e imagem reconstruída no tocante a qualidade visual, como por exemplo o índice de Similaridade Estrutural (SSIM);
- Buscar ou desenvolver outras métricas para quantificação da quantidade de informação na imagem, estimando a medida de informação e conteúdo da imagem.

REFERÊNCIAS

- ADITYA SHARMA. *Variational Autoencoder in TensorFlow*. 2021. <<https://learnopencv.com/variational-autoencoder-in-tensorflow/>>.
- AGGARWAL, C. C. et al. *Neural networks and deep learning*. [S.l.]: Springer, 2018.
- AHMED GAD. *Lossless Data Compression Using Arithmetic Encoding in Python and Its Applications in Deep Learning*. 2021. <<https://neptune.ai/blog/lossless-data-compression-using-arithmetic-encoding-in-python-and-its-applications-in-deep-learning>>.
- AHMED, N.; NATARAJAN, T.; RAO, K. R. Discrete cosine transform. *IEEE transactions on Computers*, IEEE, v. 100, n. 1, p. 90–93, 1974.
- AHN, J. et al. An optical image transmission system for deep sea creature sampling missions using autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, IEEE, v. 45, n. 2, p. 350–361, 2018.
- AKYAZI, P.; EBRAHIMI, T. Learning-based image compression using convolutional autoencoder and wavelet decomposition. In: *IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2019.
- AKYAZI, P.; EBRAHIMI, T. A new end-to-end image compression system based on convolutional neural networks. In: INTERNATIONAL SOCIETY FOR OPTICS AND PHOTONICS. *Applications of Digital Image Processing XLII*. [S.l.], 2019. v. 11137, p. 111370M.
- ALEXANDRE, D. et al. An autoencoder-based learned image compressor: Description of challenge proposal by nctu. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2019. p. 2539–2542.
- ALLOTTA, B. et al. A new auv navigation system exploiting unscented kalman filter. *Ocean Engineering*, Elsevier, v. 113, p. 121–132, 2016.
- ALT, C. V. Autonomous underwater vehicles. In: *Autonomous Underwater Lagrangian Platforms and Sensors Workshop*. [S.l.: s.n.], 2003. v. 3, p. 2.
- ANJOS, J. R. L. d. *InferenCNN: Uma biblioteca para Inferência de Redes Neurais Convolucionais Multiplataforma em OpenCL*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2019.
- ANTÔNIO, D. A. F. G. *Transreceptor acústico de comunicação digital sem fios para nó sensor subaquático*. Tese (Doutorado), 2016.

- ARABURA, G. d. S. O padrão de compressão de video mpeg moving picture experts group. Universidade Estadual Paulista (UNESP), 2012.
- BAIG, M. H.; KOLTUN, V.; TORRESANI, L. Learning to inpaint for image compression. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2017. p. 1246–1255.
- BALLÉ, J. Efficient nonlinear transforms for lossy image compression. In: IEEE. *2018 Picture Coding Symposium (PCS)*. [S.l.], 2018. p. 248–252.
- BALLÉ, J.; LAPARRA, V.; SIMONCELLI, E. P. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281*, 2016.
- BALLÉ, J.; LAPARRA, V.; SIMONCELLI, E. P. End-to-end optimization of nonlinear transform codes for perceptual quality. In: IEEE. *2016 Picture Coding Symposium (PCS)*. [S.l.], 2016. p. 1–5.
- BALLÉ, J.; LAPARRA, V.; SIMONCELLI, E. P. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.
- BALLÉ, J. et al. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.
- BÉGAINT, J. et al. Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*, 2020.
- BENGIO, Y. *Learning deep architectures for AI*. [S.l.]: Now Publishers Inc, 2009.
- BHAWNA, G. *Image compression using discrete cosine transform and discrete wavelet transform*. Tese (Doutorado), 2010.
- BOTAN, R. Implementação de algoritmo para criação de mosaico em imagens de microscopia digital de baixo custo. *Projeto de Graduação, Universidade Federal do Espírito Santo*, 2016.
- BRAND, F.; FISCHER, K.; KAUP, A. Rate-distortion optimized learning-based image compression using an adaptive hierarchical autoencoder with conditional hyperprior. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2021. p. 1885–1889.
- BROWN, M.; SZELISKI, R.; WINDER, S. Multi-image matching using multi-scale oriented patches. In: IEEE. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. [S.l.], 2005. v. 1, p. 510–517.
- BRUCKERT, A. et al. Deep saliency models: The quest for the loss function. *Neurocomputing*, Elsevier, v. 453, p. 693–704, 2021.
- CAO, K.; PENG, Y.-T.; COSMAN, P. C. Underwater image restoration using deep networks to estimate background light and scene depth. In: IEEE. *2018 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. [S.l.], 2018. p. 1–4.
- CHANG, C.-L.; GIROD, B. Direction-adaptive discrete wavelet transform for image compression. *IEEE Transactions on Image Processing*, IEEE, v. 16, n. 5, p. 1289–1302, 2007.

CHENG, Z. et al. Deep convolutional autoencoder-based lossy image compression. In: IEEE. *2018 Picture Coding Symposium (PCS)*. [S.l.], 2018. p. 253–257.

CHENG, Z. et al. Energy compaction-based image compression using convolutional autoencoder. *IEEE Transactions on Multimedia*, IEEE, v. 22, n. 4, p. 860–873, 2019.

CHENG, Z. et al. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 7939–7948.

CHOI, Y.; EL-KHAMY, M.; LEE, J. Variable rate deep image compression with a conditional autoencoder. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 3146–3154.

CHRIST, D.; WERNLI, L. Therov manual: A user guide for observation class remotely operated vehicles. 2007.

CHRISTOPOULOS, C.; SKODRAS, A.; EBRAHIMI, T. The jpeg2000 still image coding system: an overview. *IEEE transactions on consumer electronics*, IEEE, v. 46, n. 4, p. 1103–1127, 2000.

COELHO, L. d. L. V.; PAPARELLI, R. A experiência do trabalhador offshore: o caso de operadores de rov. *Proceedings of the 1nd Seminário de Saúde do Trabalhador de Franca*, SciELO Brasil, 2010.

CORREA, J. D. A. et al. Compressão de dados com perda em dispositivos da internet das coisas. *Universidade Federal de Santa Catarina*, 2020.

Data Science Academy. *Capítulo 60 – Variational Autoencoders (VAEs) – Definição, Redução de Dimensionalidade, Espaço Latente e Regularização*. 2020. (<https://www.deeplearningbook.com.br/variational-autoencoders-vaes-definicao-reducao-de-dimensao-latente-e-regularizacao/>).

Denny Ceccon. *Os tipos de redes neurais*. 2020. (<https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/>).

DIAO, E.; DING, J.; TAROKH, V. Drasic: Distributed recurrent autoencoder for scalable image compression. In: IEEE. *2020 Data Compression Conference (DCC)*. [S.l.], 2020. p. 3–12.

DONY, R. D.; HAYKIN, S. Neural network approaches to image compression. *Proceedings of the IEEE*, IEEE, v. 83, n. 2, p. 288–303, 1995.

DREWS, P. Robótica subaquática: Uma abordagem baseada em visão computacional aplicada a slam. *Trabalho de Conclusão do Curso de Graduação em Engenharia de Computação*, FURG, Rio Grande, RS, Brasil, 2008.

DUARTE, A. et al. A dataset to evaluate underwater image restoration methods. In: IEEE. *OCEANS 2016-Shanghai*. [S.l.], 2016. p. 1–6.

DUARTE, A. C. *Geração de Conjuntos de Dados para Aplicações de Visão Computacional em Ambientes Subaquáticos*. 83 f. Dissertação (Mestrado), em Engenharia de Computação — Universidade Federal do Rio Grande, Rio Grande, 2017.

DUARTE, C. M. R. *ISIAAW-Interpretação Semântica de Imagens na Avaliação da Acessibilidade Web*. Tese (Doutorado), 2018.

DUMAS, T.; ROUMY, A.; GUILLEMOT, C. Autoencoder based image compression: can the learning be quantization independent? In: IEEE. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. [S.l.], 2018. p. 1188–1192.

EUSÉBIO, P.; ASCENSO, J.; PEREIRA, F. Optimizing an image coding framework with deep learning-based pre-and post-processing. In: IEEE. *2020 28th European Signal Processing Conference (EUSIPCO)*. [S.l.], 2021. p. 506–510.

FALCAO, A. da C.; ANTUNES, R. Gerenciamento de projeto em inspeção visual subaquática com rov. *Boletim do Gerenciamento*, v. 20, n. 20, p. 25–33, 2020.

FAN, W. et al. Tapping the power of text mining. *Communications of the ACM*, ACM New York, NY, USA, v. 49, n. 9, p. 76–82, 2006.

FARR, N. et al. An integrated, underwater optical/acoustic communications system. In: IEEE. *OCEANS'10 IEEE SYDNEY*. [S.l.], 2010. p. 1–6.

FILHO, G. J. T. d. A. *Estratégias para melhoria de um CODEC de imagem baseado na transformada discreta manobrável do cosseno*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2019.

GAYA, J. F. d. O. *Processamento de imagens adquiridas em meios participativos usando deep learning*. Dissertação (Mestrado), 2018.

GHODRATI, V. et al. Mr image reconstruction using deep learning: evaluation of network structure and loss functions. *Quantitative imaging in medicine and surgery*, AME Publications, v. 9, n. 9, p. 1516, 2019.

GIODINI, S.; SPEK, E. van der; DOL, H. Underwater communications and the level of autonomy of auvs. *Retrieved on December*, v. 16, p. 2016, 2015.

GIVEKI, D.; SOLTANSHAHI, M. A.; MONTAZER, G. A. A new image feature descriptor for content based image retrieval using scale invariant feature transform and local derivative pattern. *Optik*, Elsevier, v. 131, p. 242–254, 2017.

GONZALEZ, R. C.; WOODS, R. E. *Processamento de imagens digitais*. [S.l.]: Editora Blucher, 2000.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016.

GOOGLE. *WebP: Compression techniques*. [S.l.]: Acesso em Dezembro de 2021, 2021. <<https://developers.google.com/speed/webp/docs/compression>>.

HAYKIN, S. *Neural Networks and Learning Machines - 3rd ed.* [S.l.]: Pearson, 2009.

HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

HE, Y. et al. Low bit rate underwater video image compression and coding method based on wavelet decomposition. *China Communications*, IEEE, v. 17, n. 9, p. 210–219, 2020.

HENDERI, H.; WAHYUNINGSIH, T.; RAHWANTO, E. Comparison of min-max normalization and z-score normalization in the k-nearest neighbor (knn) algorithm to test the accuracy of types of breast cancer. *International Journal of Informatics and Information Systems*, v. 4, n. 1, p. 13–20, 2021.

HOLLINGER, G. A. et al. Uncertainty-driven view planning for underwater inspection. In: IEEE. *2012 IEEE International Conference on Robotics and Automation*. [S.l.], 2012. p. 4884–4891.

HOSSEINI, M. A survey of data compression algorithms and their applications. *Network Systems Laboratory, School of Computing Science, Simon Fraser University, BC, Canada*, 2012.

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.

HU, Y. et al. Underwater image restoration based on convolutional neural network. In: *Asian Conference on Machine Learning*. [S.l.: s.n.], 2018. p. 296–311.

HU, Y. et al. Learning end-to-end lossy image compression: A benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 2021.

HUNG, E. M. Compensação de movimento utilizando módulos multi-escala e forma variável em um codec de vídeo híbrido. *Universidade de Brasília*, 2007.

HUSZAR, F. et al. Lossy image compression with compressive autoencoders. 2020.

III, A. F. H.; ZORZI, M. Modeling the underwater acoustic channel in ns2. In: *1st International ICST Workshop on Network Simulation Tools*. [S.l.: s.n.], 2010.

ILIOPOULOU, S. Deep learning based image compression - msc thesis. 2022.

INDEX, C. V. N. Cisco visual networking index: Forecast and methodology, 2016-2021. *Cisco, San Jose, CA, USA, White Paper*, 2016.

INOUE, Y. et al. Deep joint source-channel coding and modulation for underwater acoustic communication. In: IEEE. *2021 IEEE Global Communications Conference (GLOBECOM)*. [S.l.], 2021. p. 1–7.

ISI-TICs. *Aprendizagem Supervisionada ou Não Supervisionada?* 2018. <<https://isitics.com/2018/08/28/aprendizagem-supervisionada-ou-nao-supervisionada/>>.

JESUS, S.; ROCHA, W.; BITTENCOURT, J. Análise de desempenho de detectores e descritores de características utilizando a plataforma computacional raspberry pi. In: SBC. *Anais da XIX Escola Regional de Computação Bahia, Alagoas e Sergipe*. [S.l.], 2019. p. 372–381.

JIANG, F. et al. An end-to-end compression framework based on convolutional neural networks. *IEEE Transactions on Circuits and Systems for Video Technology*, IEEE, v. 28, n. 10, p. 3007–3018, 2017.

JIANG, J. Image compression with neural networks—a survey. *Signal processing: image Communication*, Elsevier, v. 14, n. 9, p. 737–760, 1999.

JIANG, Q. et al. A novel deep neural network for noise removal from underwater image. *Signal Processing: Image Communication*, Elsevier, v. 87, p. 115921, 2020.

JOHNSTON, N. et al. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 4385–4393.

JORDAN, JEREMY. *Variational Autoencoders*. 2018. (<https://www.jeremyjordan.me/variational-autoencoders/>).

JUNG, H. C. Predictive image compression using autoencoders. 2021.

KABBAI, L. et al. Image matching based on lbp and sift descriptor. In: IEEE. *2015 IEEE 12th International Multi-Conference on Systems, Signals & Devices (SSD15)*. [S.l.], 2015. p. 1–6.

KRISHNARAJ, N. et al. Deep learning model for real-time image compression in internet of underwater things (iout). *Journal of Real-Time Image Processing*, Springer, v. 17, n. 6, p. 2097–2111, 2020.

LAMÔNICA, M. T.; JÚNIOR, J. V. G. Uma perspectiva do desenvolvimento econômico da região do submédio são francisco. *Cadernos do Desenvolvimento*, p. 11, 2019.

LANBO, L.; SHENGLI, Z.; JUN-HONG, C. Prospects and problems of wireless communication for underwater sensor networks. *Wireless Communications and Mobile Computing*, Wiley Online Library, v. 8, n. 8, p. 977–994, 2008.

LANGDON, G. G. An introduction to arithmetic coding. *IBM Journal of Research and Development*, IBM, v. 28, n. 2, p. 135–149, 1984.

LAPARRA, V.; CAMPS-VALLS, G.; MALO, J. Pca gaussianization for image processing. In: IEEE. *2009 16th IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2009. p. 3985–3988.

LEÃO, J. P. C. F. et al. Comunicações rádio subaquáticas. 2012.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436–444, 2015.

LEE, J.; CHO, S.; BEACK, S.-K. Context-adaptive entropy model for end-to-end optimized image compression. *arXiv preprint arXiv:1809.10452*, 2019.

LIMA, A. S. d. et al. Control of the movement of a rov camera; controle de posicionamento da camera de um rov. 2004.

LINDEBERG, T. Scale invariant feature transform. 2012.

LIU, D. et al. Deep learning-based video coding: A review and a case study. *ACM Computing Surveys (CSUR)*, ACM New York, NY, USA, v. 53, n. 1, p. 1–35, 2020.

- LIU, Z. et al. Image compression based on octave convolution and semantic segmentation. *Knowledge-Based Systems*, Elsevier, v. 228, p. 107254, 2021.
- LLOPIS, J. S.; MARÍ, J. M.; PÉREZ-MUELAS, V. L. Clasificación de imágenes naturales y multi-espectrales mediante redes neuronales convolucionales y normalización divisiva. 2019.
- LOWE, D. G. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, Springer, v. 60, n. 2, p. 91–110, 2004.
- LOWE, G. Sift-the scale invariant feature transform. *Int. J.*, v. 2, n. 91-110, p. 2, 2004.
- LYNN, D.; BOHLANDER, G. Performing ship hull inspections using a remotely operated vehicle. In: IEEE. *Oceans' 99. MTS/IEEE. Riding the Crest into the 21st Century. Conference and Exhibition. Conference Proceedings (IEEE Cat. No. 99CH37008)*. [S.l.], 1999. v. 2, p. 555–562.
- MA, H. et al. End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, 2020.
- MA, J. et al. Image matching from handcrafted to deep features: A survey. *International Journal of Computer Vision*, Springer, v. 129, n. 1, p. 23–79, 2021.
- MACHADO, E. M. d. S. *Modem acústico subaquático a 1 Mbit/s*. Tese (Doutorado), 2016.
- MAHDI, O. A.; MOHAMMED, M. A.; MOHAMED, A. J. Implementing a novel approach an convert audio compression to text coding via hybrid technique. *International Journal of Computer Science Issues (IJCSI)*, International Journal of Computer Science Issues (IJCSI), v. 9, n. 6, p. 53, 2012.
- MARQUES, E. R. et al. Auv control and communication using underwater acoustic networks. In: IEEE. *OCEANS 2007-Europe*. [S.l.], 2007. p. 1–6.
- MENDES, ANDRÉ CRUZ e ALENCAR, MAYCOL DE. *Detalhando amostragem e quantização/Detailing sampling and quantization*. 2012. <https://maalencar.wordpress.com/2012/03/15/detailing-sampling-and-quantization/>.
- MENTZER, F. et al. Conditional probability models for deep image compression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2018. p. 4394–4402.
- MENTZER, F. et al. Practical full resolution learned lossless image compression. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 10629–10638.
- MENTZER, F. et al. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, v. 33, p. 11913–11924, 2020.
- MINNEN, D.; BALLÉ, J.; TODERICI, G. D. Joint autoregressive and hierarchical priors for learned image compression. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2018. p. 10771–10780.

MINNEN, D. et al. Image-dependent local entropy models for learned image compression. In: IEEE. *2018 25th IEEE International Conference on Image Processing (ICIP)*. [S.l.], 2018. p. 430–434.

MISHRA, D.; SINGH, S. K.; SINGH, R. K. Lossy medical image compression using residual learning-based dual autoencoder model. In: IEEE. *2020 IEEE 7th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON)*. [S.l.], 2020. p. 1–5.

MISHRA, D.; SINGH, S. K.; SINGH, R. K. Deep architectures for image compression: a critical review. *Signal Processing*, Elsevier, v. 191, p. 108346, 2022.

MUKHERJEE, D.; WU, Q. J.; WANG, G. A comparative experimental study of image feature detectors and descriptors. *Machine Vision and Applications*, Springer, v. 26, n. 4, p. 443–466, 2015.

MURPHY, C. A. *Progressively communicating rich telemetry from autonomous underwater vehicles via relays*. [S.l.], 2012.

NUNES, P. C. Auv para inspeção e mapeamento de dutos e equipamentos submarinos. *Projeto de Graduação*, 2016.

OLIVEIRA, A. de. Pré-sal: a oportunidade e seus desafios. *Revista USP*, n. 95, p. 23–32, 2012.

OLIVEIRA, H. C. de. Extração dos pontos característicos de uma imagem através do método sift. 2013.

OLIVEIRA, V. Alves de et al. Reduced-complexity end-to-end variational autoencoder for on board satellite image compression. *Remote Sensing*, Multidisciplinary Digital Publishing Institute, v. 13, n. 3, p. 447, 2021.

ORORBIA, A. G. et al. Learned neural iterative decoding for lossy image compression systems. In: IEEE. *2019 Data Compression Conference (DCC)*. [S.l.], 2019. p. 3–12.

PARMAR, C. K.; PANCHOLI, K. A review on image compression techniques. *Journal of Information. Knowledge and Research in Electrical Engineering*, v. 2, n. 2, p. 281–284, 2015.

PASSOS, BIANKA TALLITA. *O mundo do ponto de vista das Redes Neurais Convolucionais*. 2021. <<https://ateliware.com/blog/redes-neurais-convolucionais>>.

PAULO, J. V. d. Segmentação da coluna vertebral humana por meio do processamento de imagens externas da região dorsal. Universidade Estadual Paulista (UNESP), 2018.

PENNY, W. I. *Controlador dinâmico de energia para o codificador HEVC baseado em frentes de pareto*. Dissertação (Mestrado) — Universidade Federal de Pelotas, 2016.

PÉREZ, J. et al. Robotic manipulation within the underwater mission planning context. In: *Motion and Operation Planning of Robotic Systems*. [S.l.]: Springer, 2015. p. 495–522.

PINTELAS, E.; LIVIERIS, I. E.; PINTELAS, P. E. A convolutional autoencoder topology for classification in high-dimensional noisy image datasets. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 21, n. 22, p. 7731, 2021.

PINTO, W. *Modelagem dinâmica de um robô submarino semi-autônomo (tipo ROV) para inspeção de risers*. Tese (Doutorado) — Dissertação de M. Sc., COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2012.

QIAN, Y. et al. Entroformer: A transformer-based entropy model for learned image compression. *arXiv preprint arXiv:2202.05492*, 2022.

RABBANI, M. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, International Society for Optics and Photonics, v. 11, n. 2, p. 286, 2002.

RIBAS, J. P. I. F. et al. Um modelo de compressão de imagens digitais baseado em quantização vetorial e transformações afins. Universidade Federal de Uberlândia, 2008.

RIBEIRO, F. J. L. *SISTEMA DE MONITORAMENTO SUBAQUÁTICO PARA EXPLORAÇÃO DE PETRÓLEO USANDO REDES DE SENSORES ACÚSTICOS*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2012.

RIPPEL, O.; BOURDEV, L. Real-time adaptive image compression. *arXiv preprint arXiv:1705.05823*, 2017.

ROCHA, I. B. P. *Modelagem matemática de cabos umbilicais e veículos subaquáticos não tripulados*. Dissertação (Mestrado), 2013.

RODRIGUEZ, G. C. *PROPOSTA DE UM NOVO CONCEITO DE VEÍCULO AUTÔNOMO SUBAQUÁTICO*. Tese (Doutorado) — Universidade Federal do Rio de Janeiro, 2016.

RUBINO, E. M. et al. Wireless image compression and transmission for underwater robotic applications? *IFAC-PapersOnLine*, Elsevier, v. 48, n. 2, p. 288–293, 2015.

SALMAN, N. New image compression/decompression technique using arithmetic coding algorithm. *Journal of Zankoy Sulaimani*, v. 19, n. 1, p. 263–272, 2016.

SALTAREN, R.; BARROSO, A. R.; YAKRANGI, O. Robotics for seabed teleoperation: Part-1—conception and practical implementation of a hybrid seabed robot. *IEEE Access*, IEEE, v. 6, p. 60559–60569, 2018.

SARAVANAN. *Understanding Loss Functions in Machine Learning*. 2021. (<https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/>).

SENTO, A. Image compression with auto-encoder algorithm using deep neural network (dnn). In: IEEE. *2016 Management and Innovation Technology International Conference (MITicon)*. [S.l.], 2016. p. MIT–99.

SHANNON, C. E. A mathematical theory of communication. *The Bell system technical journal*, Nokia Bell Labs, v. 27, n. 3, p. 379–423, 1948.

SHELAR, P. A.; MAHALLE, P. N.; SHINDE, G. Secure data transmission in underwater sensor network: Survey and discussion. In: *Internet of Things, Smart Computing and Technology: A Roadmap Ahead*. [S.l.]: Springer, 2020. p. 323–360.

SHENG TAN. *Smooth Compression and Reconstruction of Human Portrait Image Using Convolutional Auto-Encoder*. 2018. (http://cs230.stanford.edu/projects_spring_2019/reports/18679819.pdf).

SHRAVANA, K. P.; VEENA, D. S. V. Review on lossless data compression using x-matchpro algorithm. In: IEEE. *2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*. [S.l.], 2017. p. 1095–1100.

SILVA, S. V. d. Projeto de uma arquitetura dedicada à compressão de imagens no padrão jpeg2000. 2005.

SKEIKA, E. L. et al. *Utilização de redes neurais completamente convolucionais para identificação e medição de crânios fetais*. Dissertação (Mestrado) — Universidade Tecnológica Federal do Paraná, 2019.

SONAGLIO, R. et al. Tecnologias e técnicas para inspeção de dutos e estruturas submarinas em águas rasas com auv. 2017.

STOJANOVIC, M. Acoustic (underwater) communications. *Wiley Encyclopedia of Telecommunications*, Wiley Online Library, 2003.

STOJANOVIC, M. On the relationship between capacity and distance in an underwater acoustic communication channel. *ACM SIGMOBILE Mobile Computing and Communications Review*, ACM New York, NY, USA, v. 11, n. 4, p. 34–43, 2007.

SUBRAMANYA, A. Image compression technique. *IEEE potentials*, IEEE, v. 20, n. 1, p. 19–23, 2001.

SUN, Y. et al. Image compression algorithm based on variational autoencoder. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2021. v. 2066, n. 1, p. 012008.

SUN, Y. et al. Learning a good representation with unsymmetrical auto-encoder. *Neural Computing and Applications*, Springer, v. 27, n. 5, p. 1361–1367, 2016.

SZELISKI, R. *Computer vision: algorithms and applications*. [S.l.]: Springer Nature, 2022.

TAVARES, A. Um estudo sobre a modelagem e o controle de veículos subaquáticos não tripulados. *Rio Grande*, 2003.

TEIXEIRA, G. D. Desenvolvimento de uma arquitetura de hardware de um estimador de vetores de movimento de precisão sub-pixel seguindo o padrão hevc. 2014.

TEO, K.; GOH, B.; CHAI, O. K. Fuzzy docking guidance using augmented navigation system on an auv. *IEEE journal of oceanic engineering*, IEEE, v. 40, n. 2, p. 349–361, 2014.

- THEIS, L. et al. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.
- THUM, C. Measurement of the entropy of an image with application to image focusing. *Optica Acta: International Journal of Optics*, Taylor & Francis, v. 31, n. 2, p. 203–211, 1984.
- TODERICI, G. et al. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085*, 2015.
- TODERICI, G. et al. Full resolution image compression with recurrent neural networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2017. p. 5306–5314.
- TYAGI. *Introduction to SIFT(Scale Invariant Feature Transform)*. 2019. <https://medium.com/data-breach/introduction-to-sift-scale-invariant-feature-transform-65d7f3a72d40>.
- UITTO, J. I. Global freshwater resources. In: *World Forests, Markets and Policies*. [S.l.: Springer, 2001. p. 47–58.
- UNITINS. *Compressão de Dados*. 2022. https://www2.unitins.br/BibliotecaMidia/Files/Documento/AVA_633682985082488750aula_6.pdf.
- URICK, R. J. Principles of underwater sound 3rd edition. *Peninsula Publishing Los Atlos, California*, v. 22, p. 23–24, 1983.
- VASILESCU, I. et al. Data collection, storage, and retrieval with an underwater sensor network. In: *Proceedings of the 3rd international conference on Embedded networked sensor systems*. [S.l.: s.n.], 2005. p. 154–165.
- VICERI. *Entendendo de vez a convolução: base para processamento de imagens*. 2020. <https://viceri.com.br/insights/entendendo-de-vez-a-convolucao-base-para-processamento-de-imagens/>.
- WALLACE, G. K. The jpeg still picture compression standard. *Communications of the ACM*, AcM New York, NY, USA, v. 34, n. 4, p. 30–44, 1991.
- WALLACE, G. K. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, IEEE, v. 38, n. 1, p. xviii–xxxiv, 1992.
- WEN, S. et al. Variational autoencoder based image compression with pyramidal features and context entropy model. In: *CVPR Workshops*. [S.l.: s.n.], 2019. p. 0.
- WICKRAMASINGHE, C. S.; MARINO, D. L.; MANIC, M. Resnet autoencoders for unsupervised feature learning from high-dimensional data: Deep models resistant to performance degradation. *IEEE Access*, IEEE, v. 9, p. 40511–40520, 2021.
- WITTEN, I. H.; NEAL, R. M.; CLEARY, J. G. Arithmetic coding for data compression. *Communications of the ACM*, ACM New York, NY, USA, v. 30, n. 6, p. 520–540, 1987.
- WU, C.-Y.; SINGHAL, N.; KRAHENBUHL, P. Video compression through image interpolation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 416–431.

- WU, J. et al. A comparative study of sift and its variants. *Measurement science review*, v. 13, n. 3, 2013.
- WU, Y. et al. Local shannon entropy measure with statistical tests for image randomness. *Information Sciences*, Elsevier, v. 222, p. 323–342, 2013.
- WYNN, R. B. et al. Autonomous underwater vehicles (auvs): Their past, present and future contributions to the advancement of marine geoscience. *Marine Geology*, Elsevier, v. 352, p. 451–468, 2014.
- XIAO, Y. et al. A review of object detection based on deep learning. *Multimedia Tools and Applications*, Springer, v. 79, n. 33, p. 23729–23791, 2020.
- YANG, F. et al. Variable rate deep image compression with modulated autoencoder. *IEEE Signal Processing Letters*, IEEE, v. 27, p. 331–335, 2020.
- YANG, T. Properties of underwater acoustic communication channels in shallow water. *The Journal of the Acoustical Society of America*, Acoustical Society of America, v. 131, n. 1, p. 129–145, 2012.
- YANG, Y.; XIAO, Y.; LI, T. A survey of autonomous underwater vehicle formation: performance, formation control, and communication capability. *IEEE Communications Surveys & Tutorials*, IEEE, v. 23, n. 2, p. 815–841, 2021.
- YASUKAWA, S. et al. Underwater image reconstruction using convolutional auto-encoder. In: ALIFE ROBOTICS. *Proceedings of International Conference on Artificial Life & Robotics (ICAROB2021)*. [S.l.], 2021. p. 262–265.
- YUAN, F. et al. Low bit-rate compression of underwater image based on human visual system. *Signal Processing: Image Communication*, Elsevier, v. 91, p. 116082, 2021.
- YUH, J. Design and control of autonomous underwater robots: A survey. *Autonomous Robots*, Springer, v. 8, n. 1, p. 7–24, 2000.
- ZANAJ, E. et al. Underwater wireless sensor networks: Estimation of acoustic channel in shallow water. *Applied Sciences*, MDPI, v. 10, n. 18, p. 6393, 2020.
- ZHANG, F. et al. Future trends in marine robotics [tc spotlight]. *IEEE Robotics & Automation Magazine*, IEEE, v. 22, n. 1, p. 14–122, 2015.
- ZHANG, J.; HSU, W.; LEE, M. L. Image mining: Issues, frameworks and techniques. In: UNIVERSITY OF ALBERTA. *Proceedings of the 2nd ACM SIGKDD International Workshop on Multimedia Data Mining (MDM/KDD'01)*. [S.l.], 2001.
- ZHANG, S. et al. A fast elm-based machine compression scheme for underwater image transmission on a low-bandwidth acoustic channel. *Sensor Review*, Emerald Publishing Limited, 2019.
- ZHANG, Y. A better autoencoder for image: Convolutional autoencoder. In: *ICONIP17-DCEC*. Available online: http://users.cec.sanu.edu.au/Tom.Ge-deon/conf/ABCs2018/paper/ABCs2018_paper_58.pdf. [S.l.: s.n.], 2018.

ZHOU, L. et al. Variational autoencoder for low bit-rate image compression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.: s.n.], 2018. p. 2617–2620.

ZHUANG, M. et al. A robustness and low bit-rate image compression network for underwater acoustic communication. In: SPRINGER. *International Conference on Neural Information Processing*. [S.l.], 2019. p. 106–116.