ID-SUBSEA

INNOVATION & DEVELOPMENT OF SUBSEA TECHNOLOGY

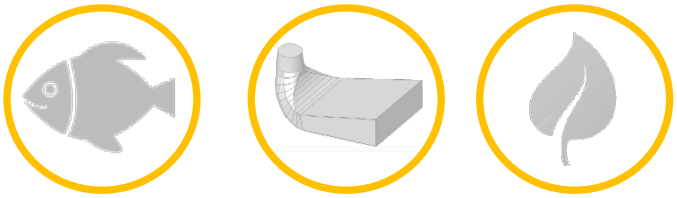
Proteção à Ictiofauna

Desenvolvimento de Técnicas de Controle e Estudos da Espécies *Pimelodus maculatus* no Tubo de Sucção a Jusante da UHE Machadinho-RS

ENGIE BRASIL ENEGRIA S.A.

ENGIE

P&D – 00403-0046/2019



Todas as informações constantes neste documento pertencem a ID-SUBSEA COM. & SERV. LTDA. Estas devem ser cedidas exclusivamente ao cliente ENGIE BRASIL ENERGIA S.A. E vedada qualquer divulgação ou circulação, sem consentimento expresso da ID-SUBSEA COM. E SERV. LTDA.

INDICE

1. CONTROLE DE REVISÃO 4
2. OBJETIVO 5
3. EQUIPE 5
4. CRONOGRAMA 6
5. ENTREGAS 9
5.1. ENTREGA 1 9
5.1.1 REVISÃO BIBLIOGRÁFICA 9
5.1.2 HARDWARE E IMAGENS DE REFERÊNCIA 10
5.1.3 MEDIÇÕES 11
5.1.4 RECURSOS UTILIZADOS 12
5.2. ENTREGA 3 12
5.2.1 BANCO DE DADOS 12
5.2.2 EXPERIMENTOS DE TRATAMENTO DE IMAGENS 13
5.2.3 RESULTADOS 16
5.2.4 CONCLUSÕES 20
5.2.5 MEDIÇÕES 21
5.2.6 RECURSOS UTILIZADOS 21
5.3. ENTREGA 4 22
5.3.1 AVALIAÇÃO DO SDK 22
5.3.2 APLICAÇÃO DE MORFOLOGIAS 22
5.3.3 APLICAÇÃO DE OPERAÇÕES MORFOLÓGICAS 25
5.3.4 MARCAÇÃO DAS IMAGENS ACÚSTICAS 27
5.3.5 RESULTADOS 28
5.3.6 CONCLUSÕES 30
5.3.7. MEDIÇÕES 33
5.3.8. RECURSOS UTILIZADOS 33
5.4. ENTREGA 5 33
5.4.1 CLASSIFICADOR LINEAR 33
5.4.2. ALGORITMOS DE RASTREAMENTO 35
5.4.3. SISTEMA SONAR 37
5.4.4. RESULTADOS 38
5.4.5. CONCLUSÕES 42

5.4.6. MEDIÇÕES 43
5.4.7. RECURSOS UTILIZADOS 43
5.5. ENTREGA 6 43
5.5.1. MEAN-SHIFT E CAMSHIFT 43
5.5.2. APRENDIZADO DE MÁQUINA E APRENDIZADO PROFUNDO 47
5.5.3. U-NET: REDES COMPLETAMENTE CONVOLUCIONAIS 48
5.5.4. ARQUITETURA YOLO 50
5.5.5. AJUSTES E RECORTES 52
6. BIBLIOGRAFIA 62

1. CONTROLE DE REVISÃO

Rev.	Ref.	Notas da Revisão
0	07/19	Elaboração do Modelo
1	07/19	JUL/19
2	12/19	DEZ/19
3	01/20	JAN//20
4	02/20	FEV/20
5	03/20	MAR/20
6	04/20	ABR/20
7	05/20	MAI/20
8	06/20	JUN/20
9	07/20	JUL/20
10	08/20	AGO/20
11	09/20	SET/20
12	11/20	NOV/20
13	12/20	DEZ/20
14	01/21	JAN/21
15	08/21	AGO/21

Tabela 1 – Revisões

2. OBJETIVO

O objetivo deste documento e detalhar tecnicamente na periodicidade mensal estipulada pelo cliente, evoluções e etapas de P&D, do algoritmo batizado de ***MARINUS_**V1*; A motivação, originalidade, aplicabilidade, relevância e razoabilidade de custos, do projeto seguem declarados no **Anexo A.**

O software por sua vez deverá ser capaz de reconhecer, distinguir e contabilizar autonomamente, peixes que entrem ou saiam da zona de controle, definida como o tubo de sucção de uma das máquinas geratrizes da UHE Machadinho-RS.

Em suma, a solução deve processar através de visão computacional, imagens acústicas, que serão geradas por sonares do tipo multifeixe, de operação em tempo real. A acurácia desta aplicação deve ser implementada e refinada por meio de técnicas de *machine learning*. O detalhamento dos princípios de treinamento do software e avaliação de sua precisão, deverão ser esmiuçados ao longo do desenvolvimento e etapas.

A equipe técnica envolvida, deve também produzir artigo científico voltado a disseminação de conhecimento no âmbito socio-acadêmico.

O projeto piloto busca sobretudo reforçar consciência eco-sustentável do *sponsor* **Engie Brasil Energia S.A**., pela iniciativa de adequar suas atividades, a processos que resguardem a vida, e biodiversidade expressa em fauna e flora.

3. EQUIPE

Nome	Função	Vinculação
João Lucas P. A. de Castro	Pesquisador	ID-SUBSEA
Renan S. Ocampo	Pesquisador	ID-SUBSEA
Wesley Peixoto	Pesquisador	INATEL
Daniel A. Vital	Orientador	UNIFESP
Mayara Cruz	Administrativo	ID-SUBSEA

Tabela 2 – Membros da equipe

4. CRONOGRAMA

Conforme estabelece o anexo A, Plano de trabalho – ID-SUBSEA,

Projeto de P&D “Proteção à Ictiofauna – Desenvolvimento de Técnicas de Controle e Estudos da Espécie Pimelodus maculatus no Tubo de Sucção e Jusante da UHE Machadinho:

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR
% Conc.	**8**	**Pesquisa e definições**																				
100	8.1	Pesquisa sobre hardware e nº de imagens de ref. a serem utilizados. Elaboração de relatório.																				
100	8.2	Definição do hardware																				
100	8.3	Definição do nº de Imagens referência																				
100	8.4	Aceitação - Gestão de Pesquisa																				
100	8.5	Revisão bibliográfica																				
100	♦	**ENTREGA 1: Relatório técnico da Etapa 8 (Faturamento: Etapa 8)**																				
	9	**Desenvolvimento do software**																				
100	9.1	Resumo geral, proposta, objetivo, técnicas a adotar. Elaboração de relatório.																				
60	9.2	Escrita do(s) algoritmo(s)																				
0	♦	**ENTREGA 2: Relatório de entrega de subetapa (Faturamento: Subetapa 9.2)**																				
100	9.3	Definição de banco de dados (imagens)																				
100	9.4	Implementação de reconhecimento de imagens por VC																				

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR
100	9.5	Aceitação - Gestão do Desenvolvimento																				
100	9.6	Aplicações de filtros																				
100	♦	**ENTREGA 3: Relatório de entrega de subetapa (Faturamento: Subetapas 9.3, 9.4, 9.5, 9.6)**																				
100	9.7	Aplicação de operadores																				
100	9.8	Aplicação de morfologias																				
100	♦	**ENTREGA 4: Relatório de entrega de subetapa (Faturamento: Subetapas 9.7 e 9.8)**																				
100	9.9	Correções																				
0	♦	**ENTREGA 5: Relatório técnico da Etapa 9 (Faturamento: Subetapas 9.1 e 9.9)**																				
	10	**Machine Learning - Aprendizado do software**																				
100	10.1	Aplicação de Haar Cascade																				
100	10.2	Criação do 1° Script																				
0	10.3	Criação do 2° Script																				
0	10.4	Criação do 3° Script																				
0	10.5	Criação do 4° Script																				
0	♦	**ENTREGA 6: Relatório de entrega de subetapa (Faturamento: Subetapas 10.1 a 10.5)**																				
0	10.6	Criação do 5° Script																				

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
			AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN	FEV	MAR
0	10.7	Criação do 6° Script																				
0	10.8	Criação do 7° Script																				
0	10.9	Aplicação da Inteligência Artificial																				
0	10.10	Medição de precisão																				
	11	**Integração hardware & software**																				
0	11.1	Instalação do software nos PCs da sala de controle																				
0	♦	**MARCO: Instalação do software na usina**																				
0	11.2	Teste de detecção																				
0	11.3	Correções																				
0	11.4	Inspeção das instalações																				
0	11.5	Entrega final																				
0	♦	**MARCO: Treinamento dos operadores**																				
0	♦	**ENTREGA 7: Relatório Final – referente ao projeto completo (software + instalação do sonar) (Faturamento: Subetapas 10.6 a 10.10, e Etapa 11)**																				

Legenda:

Previsto | Executado | Concluído | Readequado | Entregas | Marcos Relevantes | Não executado

5. ENTREGAS

Entrega	Mês	Etapa | Subetapa
1	**2**	**8**
2	**4**	**9.2**
3	**6**	**9.3, 9.4, 9.5, 9.6**
4	**10**	**9.7 e 9.8**
5	**12**	**9.9**
6	**14**	**10.1 a 10.5**
7	18	10.6 a 10.10, 11

Tabela 3 – Entregas realizadas

5.1. ENTREGA 1

As atividades previstas na primeira entrega que se estabelece, seguem detalhadas:

- Revisão bibliográfica
- Definição de hardware necessário
- Definição do nº de imagens referência

5.1.1 REVISÃO BIBLIOGRÁFICA

O início das atividades deu-se pelo levantamento bibliográfico das publicações acadêmico científicas mais recentes sobre o uso de processamento de imagens acústicas e inteligência artificial (*machine e deep learning*) aplicadas no contexto do projeto de pesquisa proposto, e no desenvolvimento do algoritmo ***MARINUS_V1***. A ocorrência de morte de peixes em barragens de hidrelétricas são recorrentes em rios brasileiros. Embora o principal fator desses eventos sejam as operações de geração e de manutenção dos componentes da hidrelétrica, há um elevado número de variáveis devido as interações entre as usinas e a ictiofauna, tornando a identificação da causa desses eventos um desafio mais complexo (Amorim et al., 2016). Segundo dados do IBAMA (Instituto Brasileiro do Meio Ambiente e dos Recursos Naturais), entre 2011 e 2016, foram mortos em média cerca de 60 toneladas de peixes anualmente em hidrelétricas brasileiras, ocasionando mais de R$ 250 milhões de reais em multas por ano (Amorim et al., 2016).

A ocorrência de eventos que levam ao óbito de peixes em represas de usinas hidrelétricas como apontadas anteriormente levam a uma série de problemas que envolvem desde o recebimento de multas ambientais, danificação de equipamentos ou mesmo a paralização da produção de energia, e a ameaça de espécies em extinção. Portanto, há uma grande demanda por ferramentas e dispositivos que auxiliem na detecção e monitoramento de peixes em canais

de fuga e tubos de sucção de hidrelétricas, possibilitando o acompanhamento e previsão de danos, além disso, permitindo a busca e utilização de mecanismos para retirar os peixes detectados. E dentre as principais ferramentas utilizadas para a detecção e monitoramento de peixes, o uso de sonares de alta resolução é o que mais se destaca. O uso de ecossondagem tem sido cada vez mais utilizada em sistemas aquáticos para monitoramento, detecção e aquisição de informações, desde identificação de substratos, à detecção e avaliação da distribuição da biota, incluindo macrófitas, zooplâncton e, particularmente, a classificação de peixes (Simmonds e MacLennan 2005). Por se tratar de uma técnica baseada no uso de sonares, é uma abordagem não intrusiva, ou seja, permite o monitoramento e contagem de organismos que estão na água sem distúrbios no ambiente, e amostrando praticamente toda a região de interesse basicamente em tempo real (Parker-Stetter et al. 2009). Porém, a presença de ruídos e perturbações no ambiente aquático devido a entrada de ar e movimentação da água, mesmo com sonares de alta resolução, podem restringir o seu uso em canais de fuga, uma vez que essas variações podem prejudicar a análise visual mesmo para operadores altamente treinados (Lucas & Baras 2000). Assim, o uso desta técnica apenas com a avaliação visual, sem nenhum tipo de processamento computacional, ainda carece de acurácia e resolução tanto para o monitoramento, em ambiente instável e turbulento dos canais a serem averiguados, quanto para detecção e contagem de espécimes. Consequentemente, faz-se necessário o uso de métodos computacionais baseados em visão computacional e aprendizado de máquina de imagens acústicas para o monitoramento, detecção e contagem de forma completamente automática do fluxo de peixes em canais e tubos de sucção em hidrelétricas.

Na literatura, autores já propuseram diferentes métodos automáticos para o monitoramento, identificação e classificação de espécies de peixes em imagens acústicas por meio de métodos de visão computacional e inteligência artificial para diferentes abordagens, como a detecção de pescado, monitoramento de ambientes e a contagem em canais de usinas nucleares e hidrelétricas. Em estudos mais recentes, Arvind e seus companheiros (Arvind et al., 2019) desenvolveram um algoritmo de detecção e contagem de peixes a partir de imagens de sonar para a área de *Pisciculture* baseando na segmentação dos peixes por uma rede neural convolucional (*Convolutional Neural Network* - CNN). Os autores realizaram uma série de filtragens nas imagens para remover ruídos de alta frequência, e em seguida realizaram o treinamento da rede utilizando a arquitetura Mask R-CNN a partir de imagens com peixes demarcados manualmente. Utilizando uma amostragem do sonar de 16 Hz, obtiveram acurácia média de 91% para a detecção e contagem dos peixes. Já French e colaboradores (French et al., 2018) avaliaram o uso de redes neurais de aprendizado profundo (*Deep Learning*) para a detecção de águas vivas em canais de fuga de usinas nucleares. Em um trabalho de acompanhamento por cerca de 2 anos, os pesquisadores desenvolveram um sistema de monitoramento integrado ao software *JellyMonitor* no sonar implementado, obtendo acurácia de detecção das águas vivas de 94%. Enquanto Zhou e companheiros (Zhou et al., 2019) desenvolveram um método de detecção de peixes em frames de vídeos de sonar em tempo real baseados em visão computacional e CNNs. Os autores realizaram uma série de operações de filtragem, realce de detalhes e detecção de objetos, desta forma, a literatura comprova que os objetivos propostos neste projeto possuem logica, e estão de acordo com que pesquisadores em todo o mundo veem atuando para a identificação e monitoramento de peixes em imagens acústicas de forma automática. Ficando no momento a nosso encargo, definir as melhores métodos e etapas de processamento das imagens e arquiteturas de redes neurais para possibilitar a contagem e classificação de imagens de peixes no tubo de sucção onde o sonar foi implantado, e conforme proposto.

5.1.2 HARDWARE E IMAGENS DE REFERÊNCIA

Na sequência, a equipe técnica trabalhou na definição da quantidade de imagens necessárias para o desenvolvimento da solução proposta, a arquitetura da rede neural que será utilizada para a classificação das imagens e contagem dos peixes e hardware necessário para o processamento. A determinação da quantidade de imagens para o desenvolvimento da solução proposta é essencial para a realização das análises e avaliação da acurácia tanto quanto a detecção quanto a contagem do número de peixes nas imagens. Considerando para esta avaliação Imagens Positivas quanto a presença de peixes, que serão cruciais para a determinação dos formatos e padrões dos mesmos, e Imagens Negativas, ou seja, sem a presença de peixes para aumentar a robustez da solução, de determinar quando não há a passagem destes pelo canal.

Nesta etapa, foram analisadas as publicações científicas apontadas anteriormente e a documentação oficial da *Open Source Computer Vision Library*, a fim de averiguar os principais fatores a serem relevados para determinação destes parâmetros. Concluiu-se que neste momento ainda não é possível definir como exatidão, o número de imagens necessárias de referência ou a técnica de *machine learning* a adotar para o classificador. Considera-se, portanto, estas variáveis como empíricas, a serem definidas nas próximas etapas.

Contudo e para início dos trabalhos, estabeleceu-se três testes iniciais que considerarão o número de imagens a seguir, para análise de resultados preliminares.

Testes	Imagens positivas	Imagens Negativas
Nível 1	4000	2000
Nível 2	8000	4000
Nível 3	16000	8000

Tabela 4 – Nº de imagens de referência

Estipulou-se Também que o processamento de imagens acústicas em tempo real, assim como os métodos associados de visão computacional e *machine learning*, demandariam minimamente do hardware indicado abaixo:

Processador	Intel i7-8700
Placa de vídeo	Quadro P1000
Mem. RAM	16GB DDR4
HD	1TB

Tabela 5 – Configuração do hardware de processamento

5.1.3 MEDIÇÕES

Considera a alocação dos recursos conforme explicitado abaixo, por atividade, nesta etapa (1):

Etapa	Atividade	Custo (R$)
8.1	Pesquisa Hardware e Nº de imagens de referência	1.209,96
8.2	Definição do hardware	2.149,68

8.3	Definição do Nº de imagens de referência	4.299,36
8.4	Aceitação	432,45
8.5	Revisão bibliográfica	854,72
	TOTAL	**8.946,17**

Tabela 6 – Recursos entrega 1

5.1.4 RECURSOS UTILIZADOS

Dos recursos disponíveis e elencados no Anexo A da minuta contratual, foram empregados nesta etapa (1), os detalhados abaixo:

Mês/Ano	Mat. De consumo	Mat. Perm e Equip.	Outros	Serv. 3os	Viagens e diárias	Total Mês
Jun / 19	0,00	0,00	0,00	0,00	0,00	0,00
Jul / 19	0,00	0,00	142,14	0,00	1.001,39	1.143,53

Tabela 7 – Reembolsos Entrega 1

5.2. ENTREGA 3

As atividades previstas na entrega (3) que se estabelece, seguem detalhadas:

- Definição do banco de dados;
- Implementação de reconhecimento imagens por VC;
- Aplicação de filtros;

5.2.1 BANCO DE DADOS

A primeira atividade consistia em definir melhor banco de dados para o projeto, e melhor forma de armazenamento das imagens no banco de dados em questão. Para tal definiu-se o MySQL como solução a nível banco de dados.

Os motivos que pelos quais optou-se pelo MySQL, seguem elencadas abaixo:

- Consistência
- Alta performance
- Confiabilidade
- Facilidade de implementação

- Condição open-source
- Não relevância de custos

Como metodologia para armazenamento das imagens, decidiu-se adotar o formato blob nativo do MySQL. O formato considera manutenção de dados binários, podendo estes ser imagens, arquivos de diferentes extensões, além de poder armazenar vídeos, dentre outros formatos. A principal vantagem de escolher esse formato é a de que se evite o principal problema vivido em armazenamento de imagens, a sobrecarga de sistema durante a busca por dados específicos.

5.2.2 EXPERIMENTOS DE TRATAMENTO DE IMAGENS

5.2.2.1 Materiais

Após a conclusão da instalação dos sonares foi possível iniciar a coleta de imagens necessárias para os experimentos de tratamento de imagem, como filtragem e realce de informações, que permitiram início do desenvolvimento da solução proposta. Já foram coletados aproximadamente 800GB de dados, organizados em versões brutas, e pré-processadas pelo software original do sonar. Serão avaliados para o uso neste projeto, ambos os dados, onde há e se averiguar se os filtros do software do sonar contribuem para o monitoramento e detecção dos peixes. Além disso, parte dos dados está sendo convertida para vídeo a fim de serem utilizados nos experimentos desta fase inicial, onde ocorrerão testes de visão computacional que permitam a análise dos benefícios da manipulação de imagens estáticas (frame a frame), em comparação ao tratamento de vídeos.

Tratando da parte computacional, decidiu-se pela utilização da biblioteca OpenCV para o reconhecimento e processamento das imagens acústicas, operando-a a princípio em linguagem computacional Python. OpenCV é uma biblioteca open source de visão computacional e aprendizado de máquina, possuindo uma infraestrutura comum para aplicações em visão computacional, e mais de 2500 algoritmos que incluem implementações clássicas e o estado da arte da visão computacional e aprendizado de máquina. Além disso, trata-se de uma biblioteca com amplas aplicações, disponível para diversas linguagens programacionais, como Python, C++ e C Sharp.

5.2.2.2 Métodos

Após as definições quanto ao banco de dados, bibliotecas e a linguagem programacional e para início do desenvolvimento da solução, foram realizados experimentos de aplicação de filtros para tratamento das imagens acústicas, já obtidas em campo. O propósito desta etapa, conhecida corriqueiramente como pré-processamento, é avaliar o condicionamento das imagens geradas, por meio de filtros, a fim de avaliar resultados com foco em corrigir, suavizar ou realçar determinadas regiões da imagem. As técnicas de filtragem são transformações da imagem pixel a pixel, que não dependem exclusivamente do valor de um determinado pixel, mas também dos valores dos pixels vizinhos (Clark et al., 2007) O processo de filtragem é feito utilizando matrizes denominadas máscaras (ou *Kernels* em inglês), as quais são aplicadas sobre a imagem em um processo de “varredura” por toda a imagem, onde ocorre a substituição do valor de cada pixel

por um novo valor que depende dos valores dos pixels vizinhos e dos pesos da máscara. Gerando uma nova imagem com a eliminação de ruídos e elementos indesejados para análise da mesma (Clark et al., 2007). O uso de filtros é justamente recomendado para imagens com ruídos ou perturbações, como no caso das imagens acústicas, e são utilizados para suavizar contornos, realizar correções de objetos, e afiar bordas das estruturas presentes nas imagens. Assim, os primeiros filtros a serem testados e avaliados foram o filtro média (Figura 1 (c)), gaussiano (Figura 1 (d)), mediana (Figura 1 (e)) e bilateral (Figura 1 (f)). Estes filtros são conhecidos como filtros espaciais, realizando alteração dos pixels da imagem a partir de seus vizinhos, preservando as bordas das estruturas e realizando a suavização de ruídos de alta frequência apresentados em trabalhos semelhantes da literatura com imagens acústicas (Zhou et al., 2019).



Figura 1 - Demonstração de F. Espaciais

A Figura 1 traz, (a) Imagem original. (b) Imagem Original com adição de Ruído. (c) Filtragem da Imagem Ruidosa com Filtro Média kernel 5x5. (d) Filtragem da Imagem Ruidosa com Filtro Gaussiano kernel 5x5. (e) Filtragem da Imagem Ruidosa com Filtro Mediana com kernel 5x5. e (f) Filtragem da Imagem Ruidosa com Filtro Bilateral kernel 5x5.

Filtro Média

O filtro média tem o objetivo de remover ruídos da imagem, realizando a suavização dos pixels pela média dos valores entre seus vizinhos. Como ruído podemos entender pixels com valores muito discrepantes daqueles de sua vizinhança, como apresentado na Figura 1, onde no item (a) possuímos a imagem original, e no item (b) a imagem original com a aplicação de ruído, perdendo detalhes da imagem, e havendo discrepância entre os pixels. O filtro médio considera os valores dos pixels da vizinhança, e faz uma média dos valores desses pixels, inclusive do pixel central do kernel que está sendo processado. Se usarmos uma máscara de 9 pixels (3x3) sobre o pixel sendo processado, cada pixel da vizinhança contribuirá com 1/9 do seu valor para o novo valor do pixel central. Ou seja, é realizada a média entre o pixel central, e os seus 8 vizinhos, e o valor da média substitui o valor do pixel central. Na Figura 1 (c) temos a aplicação

do filtro média na imagem ruidosa (Figura 1 (b)) utilizando uma máscara de 25 pixels (5x5), e onde é possível perceber a suavização do ruído na imagem.

Filtro Gaussiano

O Filtro Gaussiano é um operador de convolução2D que é utilizado para suavizar, as imagens e remover ruídos de alta frequência, e ao ser utilizado normalmente deixa a imagem um pouco "embaçada". É altamente recomendado para filtrar ruídos gaussianos, e a sua máscara, ao invés da máscara do filtro média que possui pesos iguais, trata-se de uma distribuição gaussiana 2D, onde o pixel central recebe o valor mais alto, e os pixels da máscara mais adjacentes recebem os menores valores (Figura 2). Na figura 1 (d) é possível analisar a aplicação do filtro gaussiano na imagem ruidosa utilizando uma máscara 5x5, onde é possível notar que como o filtro média, removeu o ruído de alta frequência, mas ao "embaçar" a imagem, perdi um pouco dos contornos da imagem.

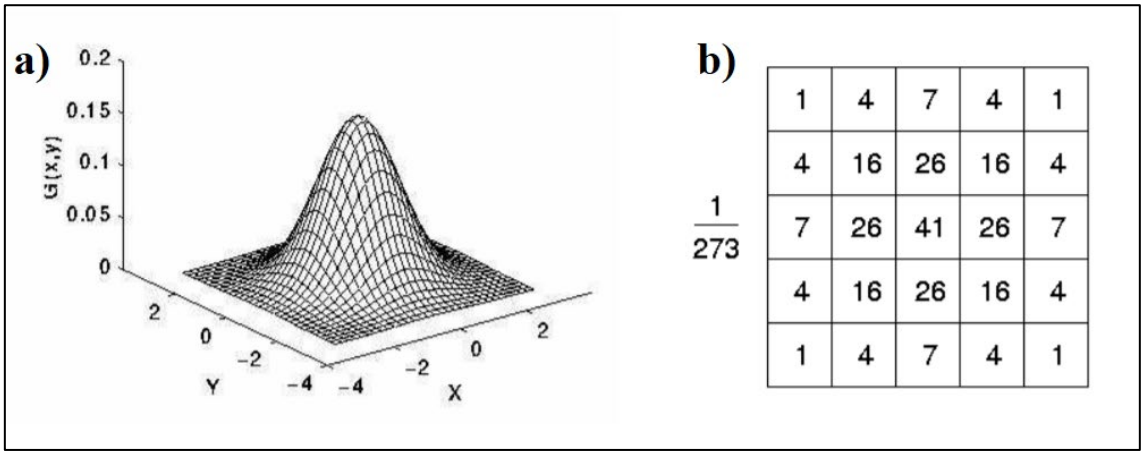


Figura 2 - Demonstração máscara filtro Gaussiano

A Figura 2 traz a demonstração da máscara filtro gaussiano. (a) Máscara na visão "2D", demostrando que os pesos dentro do kernel possuem distribuição gaussiana. (b) A Máscara 5x5 com os pesos devidamente apresentada, que passará por normalização após a divisão pelo valor ao lado que corresponde a soma dos pesos.

Filtro Mediana

O Filtro da mediana é um método bastante utilizado para suavizar ruídos do tipo impulsivo em sinais e imagens digitais conhecidos como "*Salt and Pepper*", ou seja, é responsável por retirar ruídos pontuais com alta discrepância na região. A máscara do filtro mediana não possui valores fixos, ao invés disso, ela recebe durante a varredura da imagem os próprios valores dos pixels centrais e seus vizinhos, substituindo o valor do pixel central pelo valor da mediana dos pixels na máscara. Na figura 1 (e) é possível analisar a aplicação do filtro mediana na imagem ruidosa utilizando uma máscara 5x5, apresentando que o filtro mediana consegue remover os ruídos pontuais, mas sem prejudicar as bordas da imagem.

Filtro Bilateral

O filtro bilateral é um filtro de suavização não linear com redução de ruído das imagens, mas tendo como característica principal a preservação de bordas e contornos da imagem. Ele substitui a intensidade de cada pixel por uma média ponderada dos valores de intensidade dos

pixels vizinhos, onde o peso de cada pixel da máscara pode ser baseado em uma distribuição gaussiana. Assim, ao contrário de filtro média que possui pesos iguais para toda a máscara, o que diminui o contraste nos contornos, o filtro bilateral enaltece a diferença nas fronteiras entre regiões, preservando as bordas afiadas. Na figura 1 (f) é possível analisar a aplicação do filtro bilateral na imagem ruidosa utilizando uma máscara 5x5, onde é possível analisar que o filtro é capaz de remover a maior parte do ruído preservando o alto contraste nas regiões de borda.

5.2.3 RESULTADOS

Os filtros apresentados na sessão anterior foram aplicados a imagens acústicas obtidas diretamente do sonar. O filtro de média foi o primeiro filtro testado, e este não obteve um resultado satisfatório, uma vez que suprime todo o conteúdo de alta frequência da imagem, ou seja, suaviza toda a imagem inclusive as bordas como pode ser visto na imagem abaixo.

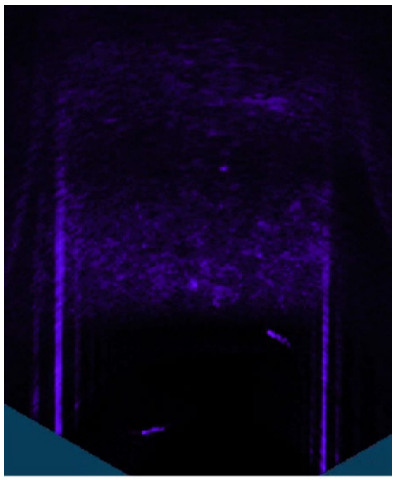


Figura 3 - Imagem original

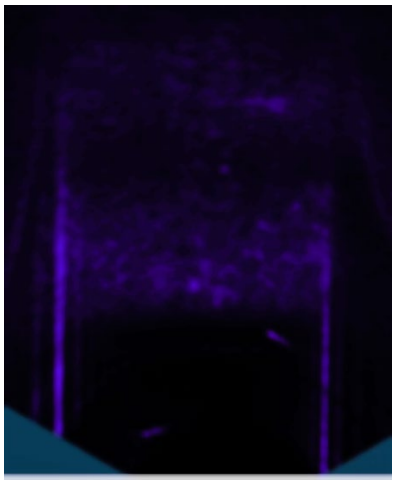


Figura 4 - Filtro de média

O filtro gaussiano assim como o filtro de média suprimi todo o conteúdo de alta frequência, sendo altamente efetivo. Para tratamento de ruídos do tipo gaussiano, frequentes em imagens acústicas de sonares. Tornando-se um método viável para as aplicações do projeto, considerando aplicação subsequente de operadores morfológicos para a detecção de estruturas e objetos desejados.

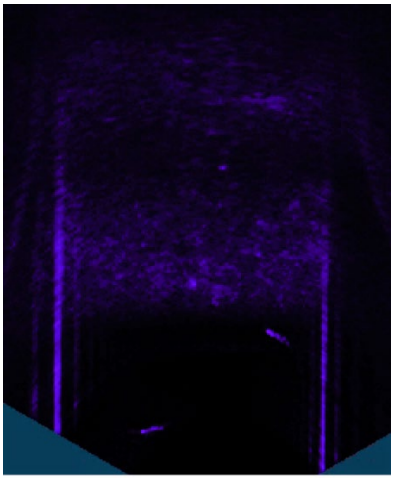


Figura 5 - Imagem original

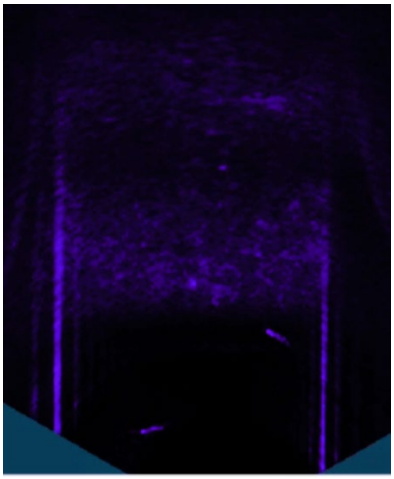


Figura 6 - Filtro Gaussiano

Já o filtro de mediana apresentou melhores resultados, quando comparado ao filtro de média e o gaussiano. Ele suaviza a imagem, retirando pixels discrepantes de alta frequência, porém preservando o conteúdo das bordas, que permanecem nítidas.

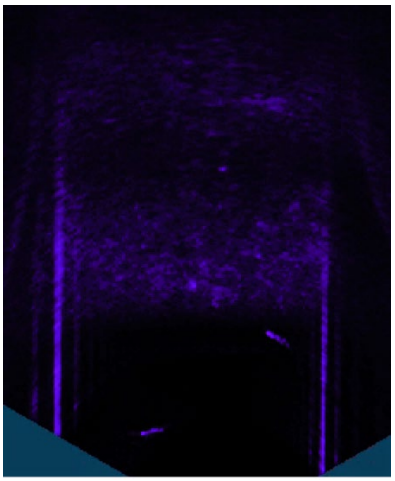


Figura 7 - Imagem original

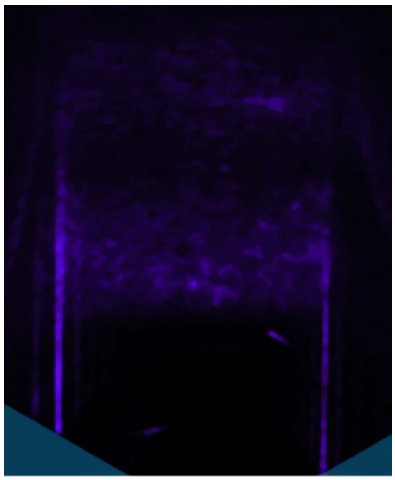


Figura 8 - Filtro de mediana

E por fim o filtro bilateral, que apresentou os melhores resultados visuais quando comparado aos demais avaliados, onde realizou a suavização da imagem, mas preservando as bordas e contornos.

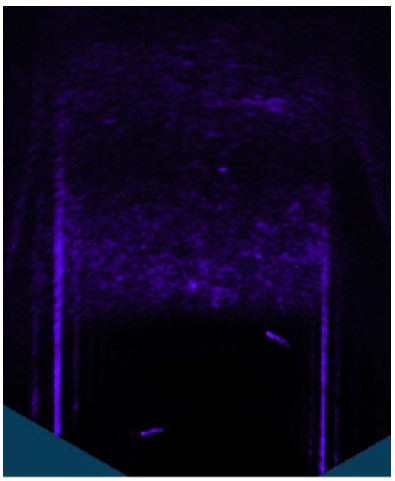


Figura 9 - Imagem original

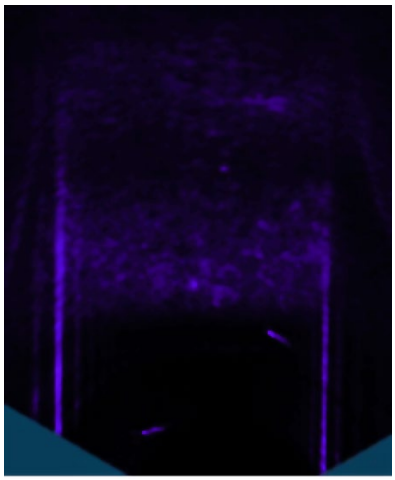


Figura 10 - Filtro Bilateral

Outra forma de avaliar a melhora da imagem após a realização da filtragem é através da avaliação da mudança do histograma desta. O histograma de uma imagem consiste na distribuição dos valores de cada pixel de uma imagem conforme apresentado na Figura 3. O eixo horizontal consiste nos valores que os pixels podem assumir, que no caso de imagens de 8 bits são 256 valores possíveis, situando-se entre 0 (preto) e 255 (branco), havendo diversos tons de cinza entre as extremidades. Já o eixo vertical apresenta o número de ocorrências do valor de cada pixel.

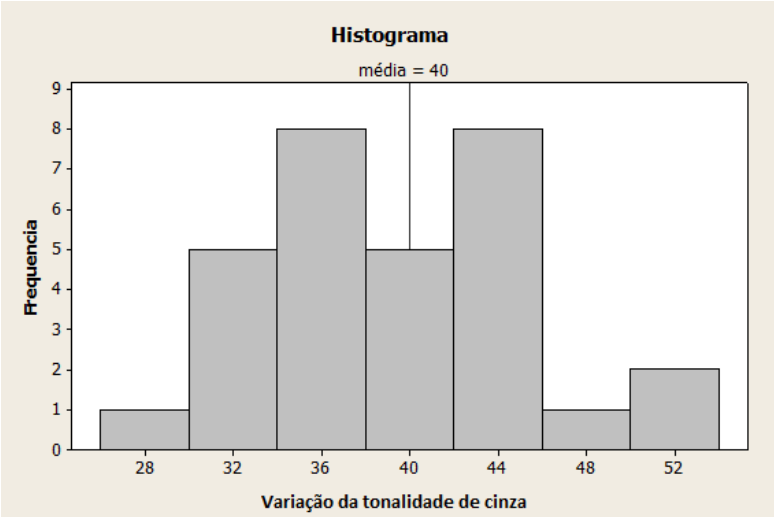


Figura 11 - Histograma

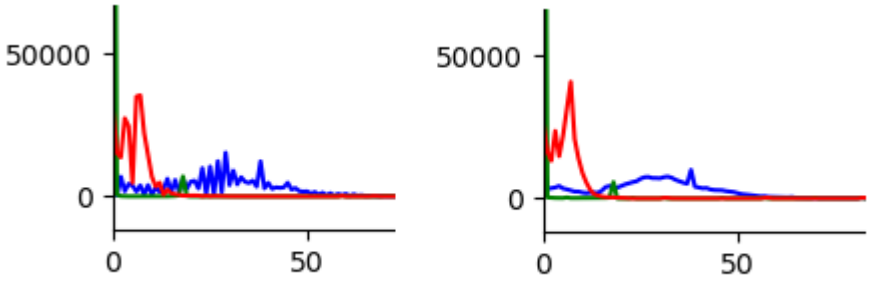


Figura 12 – Imagem Original (esquerda) x Imagem c/ filtro de média (direita).

Pode-se observar que as variações de amplitudes foram reduzidas, ou seja, diminuiu o número de pixels repetidos, inclusive aqueles que representam as bordas da imagem, o que explica a inefetividade do mesmo para o objetivo deste trabalho.

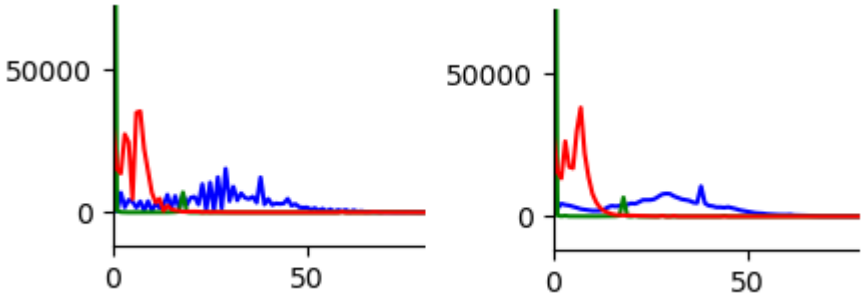


Figura 13 – Imagem Original (esquerda) x Imagem c/ filtro de gaussiano (direita).

O filtro gaussiano assim como o filtro de média reduz as variações de amplitude, fazendo com que as bordas também sejam suprimidas. Porém em imagens de sonares é bem comum ruídos do tipo gaussiano. Assim torna-se interessante utilizar este filtro, mesmo que posteriormente ainda se faça necessário a aplicação de um operador morfológico para recuperar o conteúdo das bordas suprimido.

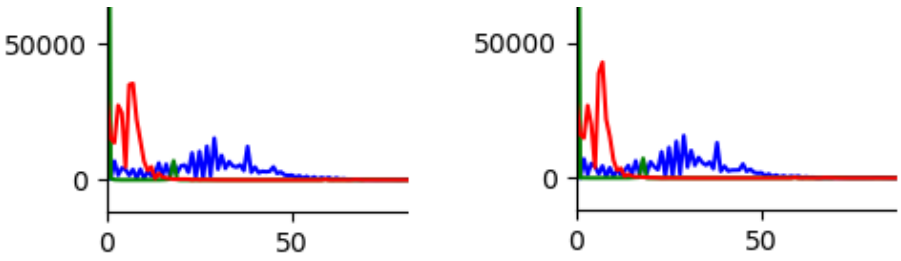


Figura 14 – Imagem Original (esquerda) x Imagem c/ filtro de mediana (direita).

O filtro de mediana reduz as variações de amplitude, porém não suprime o conteúdo de alta frequência das bordas, o que indica um melhor resultado, quando comparado ao filtro de média e o gaussiano.

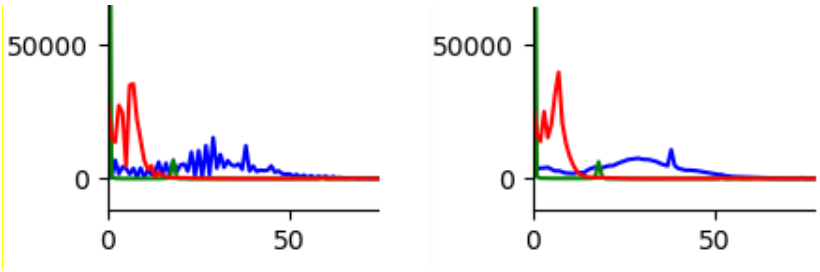


Figura 15 – Imagem Original (esquerda) x Imagem c/ filtro bilateral (direita).

Enfim o filtro bilateral, este por sua vez, suprime todo conteúdo de alta frequência preservando todo conteúdo das bordas. Visivelmente seu gráfico de histograma é o mais difícil de perceber a preservação do conteúdo das bordas, mas isso acontece devido a representação de amplitudes muito pequenas para a escala destes gráficos.

Tipo de filtro	Resultados
Média	Inefetivo
Mediana	Pouco Efetivo
Gaussiano	Inefetivo para suavização da imagem, efetivo para tratamento de ruído gaussiano.
Bilateral	Efetivo

Tabela 8 – Resumo dos resultados.

5.2.4 CONCLUSÕES

A partir das análises visuais, e avaliação das alterações do histograma para todas as filtragens realizadas, concluímos que nesta primeira etapa o filtro bilateral apresentou os melhores resultados quando a suavização e remoção de ruídos na imagem, mantendo o afiamento nas bordas entre as estruturas. O contraste entre o contorno e as regiões da imagem devem ser preservados principalmente para etapas futuras, onde será realizado a identificação e segmentação dos peixes nas imagens do sonar. A análise visual e disposição do histograma da imagem filtrada pelo filtro bilateral apresentou conciliação, com eliminação de ruídos presentes nas imagens, e preservação das bordas conforme o desejado para a aplicação.

Contudo, ainda serão avaliados filtros no domínio da frequência, além de técnicas de realce de imagem como através de ondas Wavelets, a fim de melhorar a qualidade das imagens e a distinção dos objetos de interesse. E os testes passarão a ser realizados com as imagens "brutas" capturadas diretamente do sonar, sem filtragem padrão do software original, para aplicação de filtragem definida pela equipe, e início do uso de técnicas de aprendizado de máquina. Desta forma, os próximos passos de desenvolvimento da solução proposta, tratam da avaliação das funcionalidades do SDK do sonar, para incremento das funções de acesso e tratamento das imagens brutas.

5.2.5 MEDIÇÕES

Etapa	Atividade	Custo (R$)
9.3	Definição do banco de dados	10.748,40
9.4	Implementação de Rec. de imagens por VC	21.946,80
9.5	Aceitação – Gestão de Desenv.	384,40
9.6	Aplicação de filtros	26.450,80
	TOTAL	**59.080,40**

Tabela 9 – Recursos entrega 3

5.2.6 RECURSOS UTILIZADOS

Dos recursos disponíveis e elencados no Anexo A da minuta contratual, foram empregados nesta etapa (3), os detalhados abaixo:

Mês/Ano	Mat. De consumo	Mat. Perm e Equip.	Outros	Serv. 3os	Viagens e diárias	Total Mês
Ago / 19	0,00	0,00	0,00	0,00	0,00	0,00
Set / 19	0,00	0,00	250,63	0,00	1.765,69	2.016,32
Out / 19	0,00	0,00	0,00	0,00	0,00	0,00
Nov / 19	0,00	0,00	91,67	0,00	645,82	737,49
Dez / 19	0,00	0,00	0,00	0,00	0,00	0,00
Jan / 20	0,00	0,00	0,00	0,00	0,00	0,00
Fev / 20	0,00	0,00	996,97	0,00	6.083,79	7.080,86

Tabela 10 – Reembolsos Entrega 3

5.3. ENTREGA 4

As atividades previstas na entrega (4), seguem detalhadas abaixo:

- Aplicação de operadores
- Aplicação Morfologias

5.3.1 AVALIAÇÃO DO SDK

Na vigência da entrega 4, analisou-se o SDK do software sonar, cedido pelo do fabricante. O termo SDK significa *Software Development Kit*, e consiste no conjunto de ferramentas de desenvolvimento de um software específico. No caso do *JellyMonitor*, software do sonar implementado para esta solução. A versão SDK da plataforma, permite que realizemos implementações, ajustes e adições nas próprias ferramentas nativas do sonar. O objetivo neste ponto do desenvolvimento da solução, foi a de se associar todos os métodos e soluções do algoritmo em desenvolvimento ao software *JellyMonitor,* a fim de aproveitarmos o *frontend* do software de fábrica, e otimizar o processo de obtenção das imagens cruas, sem nenhum processamento digital, oriundas do sonar. Inclusive, a ideia de adicionar a solução de monitoramento e contagem de peixes ao software nativo, otimizara o seu uso por parte daqueles que a operarem, pois irão efetivamente monitorar peixes, através de uma única aplicação.

Os testes e estudos realizados confirmaram a viabilidade de aquisição de imagens cruas oriundas do sonar, além da possibilidade de se desativar os filtros digitais disponibilizados pelo fabricante. Foi analisado ainda, o funcionamento das interações entre ferramentas e telas do *frontend*, disponíveis ao usuário, pela interface original. Verificou-se que a parte de processamento do SDK foi desenvolvida na linguagem C++, concluindo-se que a solução de monitoramento e contagem de peixes deverá ser produzida nesta mesma linguagem, a fim de agilizar a integração com os sistemas do software. Em adicional, tem-se agora ciência de que é possível realizar alterações e adições no software *JellyMonitor*, de funções necessárias a solução em desenvolvimento, o que promove otimização de tempo nas etapas do desenvolvimento do *frontend* idealizado para o produto. A aquisição de imagens brutas, pelo uso do SDK, permitiu a aplicação de filtros para realce de dados específicos, e o uso de métodos de morfologia binaria para identificação e segmentação de peixes nestas imagens.

5.3.2 APLICAÇÃO DE MORFOLOGIAS

As atividades realizadas nesta etapa buscaram trazer maior nitidez às imagens previamente filtradas, visando evidenciar ainda mais os elementos de interesse. O delineamento dos contornos que definem os peixes nas imagens foi a estratégia utilizada para realçar esses elementos, e baseou-se na utilização de operadores derivativos de primeira e

segunda ordem. Os resultados obtidos, operacionalizados com as imagens originais, permitiu que o tratamento de afiamento de bordas fosse realizado, dando origem a imagens com maior nitidez.

5.3.2.1 Máscaras de Nitidez - Identificação de bordas

A identificação das bordas que definem os objetos presentes nas imagens é essencial para o correto mapeamento, e posterior quantificação, dos elementos de interesse configurados, neste caso, como os peixes. As bordas de uma imagem podem ser caracterizadas como as regiões onde há rápida transição de intensidade entre os pixels e devem ser preservadas, e até mesmo evidenciadas, de forma que os elementos presentes na imagem sejam nítidos. Assim, a etapa de filtragem realizada anteriormente, que promove a remoção dos ruídos presentes nas imagens devido a forma de captação do sinal, pode causar a atenuação dos contornos presentes nas imagens. Esse efeito indesejado pode ser minimizado a partir da aplicação de técnicas de aguçamento que se baseiam na diferenciação espacial da imagem. Diferentes operadores derivativos, que podem ser de primeira ou segunda ordem, podem ser aplicados na tarefa de identificação de contornos e foram testados nesse projeto o operador gradiente de Sobel e o operador Laplaciano.

5.3.2.1.1 Operador Gradiente de Sobel

Os operadores derivativos de primeira ordem são aqueles que identificam toda a extensão das bordas, excluindo somente as regiões da imagem onde as intensidades dos pixels são constantes. De maneira geral, as derivações de primeira ordem consistem na diferença entre o pixel de interesse e o pixel imediatamente adjacente. O operador derivativo de primeira ordem mais comum é o gradiente, que representa um vetor que indica a direção onde ocorre a maior taxa de variação de intensidade. A construção do vetor gradiente engloba a derivação da imagem em suas duas dimensões, sendo que a magnitude do gradiente é dada pelo módulo do vetor. Algumas máscaras podem ser utilizadas para determinar as derivações de primeira ordem nas dimensões de uma imagem, como é o caso do operador de Sobel (Figura 13). Neste operador, a diferença entre as linhas da máscara Wx realiza a aproximação da derivada na direção x da imagem, dando ainda maior importância ao valor central da região. Uma relação semelhante ocorre na direção y, porém considerando as colunas da máscara Wy.

a)

-1	-2	-1
0	0	0
1	2	1

Wx

b)

-1	0	1
-2	0	2
-1	0	1

Wy

Figura 16 *– Máscaras do operador gradiente de Sobel. a) Wx responsável pela aproximação da derivação da imagem na direção X. b) Wy responsável pela aproximação da derivação da imagem na direção Y.*

5.3.2.1.2 Operador Laplaciano

O operador Laplaciano se caracteriza como um operador derivativo de segunda ordem, onde as extremidades internas e externas das bordas presentes na imagem são identificadas. Diferentemente dos operadores derivativos de primeira ordem, onde toda a transição de intensidades é registrada, os operadores derivativos de segunda ordem captam somente o início e o fim das transições de intensidade que caracterizam as bordas. Assim, como resultado obtido dessa operação, as bordas são representadas por contornos duplos de forma que os detalhes finos da imagem são realçados. As máscaras utilizadas como operadores Laplacianos possuem a característica de serem invariantes a rotação, e são conhecidas como operadores isotrópicos. Os operadores Laplacianos podem ter duas formas, sendo que a primeira leva em consideração somente as variações imediatamente adjacentes ao pixel (Figura 14 A) e a outra, além dos pixels adjacentes, considera também as variações diagonais da região (Figura 14 B). Cada um dos operadores apresentados possuem uma segunda versão, onde cada elemento da máscara é multiplicado por -1, e o resultado obtido será semelhante às máscaras convencionais.

a)

0	1	0
1	-4	1
0	1	0

b)

1	1	1
1	-8	1
1	1	1

Figura 17 *– Máscaras do operador Laplaciano. a) Operador onde somente os pixels adjacentes sejam considerados. b) Operador onde todos os pixels da região, adjacentes e diagonais, são levados em consideração.*

5.3.2.2 Afinamento de borda

A utilização dos operadores derivativos nas imagens, como descrito anteriormente, dá origem a uma nova imagem com os contornos que definem os objetos presentes. Entretanto, os contornos por si só nem sempre dão as informações buscadas na imagem e, assim, torna-se necessário uma nova operação entre a imagem original e suas bordas. Esse procedimento é conhecido como afiamento de bordas, ou também aguçamento de bordas, e nele é realizada a adição da imagem de bordas à imagem original (Figura 15). Desta forma, a imagem obtida apresenta as características iniciais, porém com realce das bordas dos objetos, fazendo com que os elementos se tornem mais nítidos, evidentes e distinguíveis entre si.

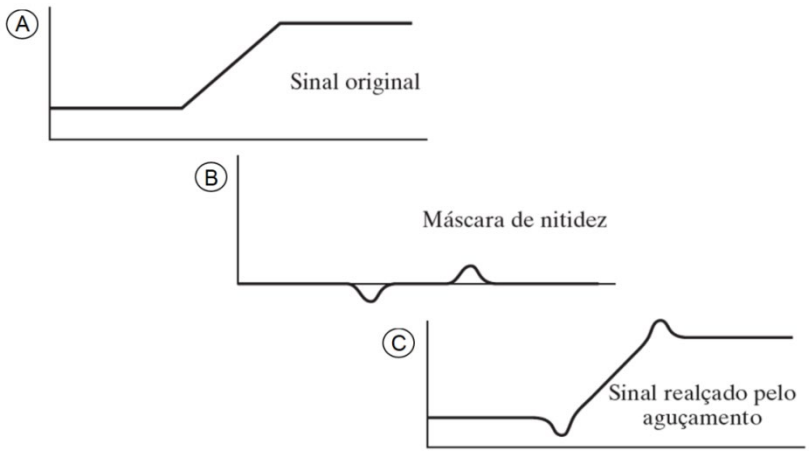


Figura 18 *– Etapas do procedimento de afiamento de bordas. A) Imagem inicial após o processo de filtragem. B) Extração das bordas da imagem através de operador derivativo. C) Adição de B a A, obtendo maior realce dos objetos presentes na imagem.*

5.3.3 APLICAÇÃO DE OPERAÇÕES MORFOLÓGICAS

O processo de afiamento de bordas possibilita que os peixes presentes nas imagens sejam evidenciados, porém a estrutura que define esses elementos pode apresentar diferentes falhas tanto em formato quanto em preenchimento. A correção dessas irregularidades deverá ser implementada com a aplicação de operações morfológicas que promovem as modificações estruturais. A extração dos objetos nas imagens tratadas até o momento, isto é, o isolamento dos peixes presentes em cada *frame* do arquivo, foi realizada através da aplicação de um conjunto de operações morfológicas. A morfologia matemática busca utilizar um elemento estruturante para promover alterações estruturais nos objetos a fim de obter correções que tornem as características buscadas mais evidentes. Os elementos estruturantes podem ser definidos como um conjunto de pixels, dispostos em formato pré-definido, que irão percorrer toda a imagem e induzir modificações. A forma de interação entre o elemento estruturante e os objetos na imagem definem o tipo de operação morfológica, sendo as principais denominadas como erosão e dilatação. Quando combinadas, as operações de abertura e fechamento são originadas. Assim, devido a relação intrínseca entre o elemento estruturante e a operação realizada, sua dimensão e formato possuem grande influência no resultado a ser obtido. A seguir, será descrito o funcionamento das operações de erosão, dilatação, abertura e fechamento.

5.3.3.1. Erosão

A operação de erosão possui o intuito de reduzir a dimensão dos objetos da imagem, de forma que se pode observar uma diminuição ou afinamento nas estruturas presentes na imagem. Isso ocorre, pois, o pixel originalmente pertencente ao objeto só será mantido caso o elemento estruturante esteja completamente englobado pela estrutura. Assim, com base no pixel central do elemento estruturante, o posicionamento deste com relação ao objeto pode ser observado e, desta forma, é decidido se o pixel alinhado será mantido ou eliminado.

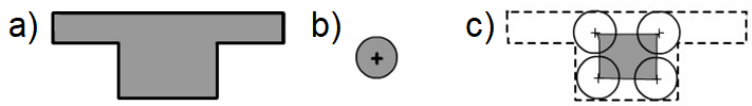


***Figura 19** – Operação morfológica de erosão. a) Objeto original. b) Elemento estruturante em forma de disco e com centro demarcado. c) Resultado da erosão de a) através de b). Fonte: Adaptação de MACHADO (2020).*

5.3.3.2. Dilatação

A operação de dilatação, em contraste com a erosão, resulta em um aumento ou engrossamento da dimensão do objeto da imagem. Enquanto a erosão elimina pixels da imagem quando todo o elemento estruturante não está englobado pelo objeto, na dilatação quando o centro do elemento estruturante se encontra alinhado a um pixel do objeto, o formato sobressalente é acrescentado à estrutura do objeto resultante. Desta forma, as dimensões e o formato do elemento estruturante irão interferir diretamente na espessura do aumento no objeto e também em sua característica estrutural.

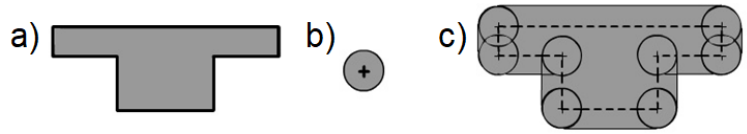


Figura 17 – Operação morfológica de dilatação. a) Objeto original. b) Elemento estruturante em forma de disco e com centro demarcado. c) Resultado da dilatação de a) através de b). Fonte: Adaptação de MACHADO (2020).

5.3.3.3. Abertura

A operação de abertura consiste em um processamento derivado tanto da erosão quanto da dilatação, uma vez que uma combinação dessas operações básicas é realizada. A primeira etapa da abertura consiste em uma aplicação de erosão na imagem e, o resultado obtido é submetido à dilatação na segunda etapa. O elemento estruturante utilizado em ambas as etapas deve ser o mesmo. Desta forma, com a operação de abertura os contornos dos objetos tendem a ser suavizados, bem como as saliências finas tendem a ser eliminadas e as regiões com dimensão menor que o elemento estruturante tendem a ser separadas.

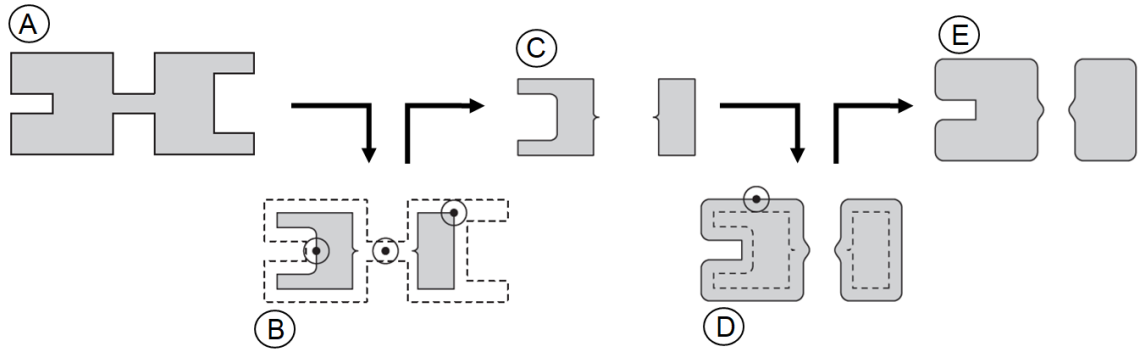


***Figura 20** – Etapas da operação de abertura de um objeto. A) Objeto original. B) Processo de erosão, usando um elemento estruturante em forma de disco. C) Resultado obtido após a erosão. D) Processo de dilatação em C, utilizando o mesmo elemento estruturante. E) Objeto após a dilatação e representando o resultado da operação de abertura.*

5.3.3.4. Fechamento

A operação de fechamento se assemelha à operação de abertura, sendo também uma composição das operações de erosão e dilatação. Entretanto, a ordem de aplicação é inversa à abertura e, portanto, inicialmente a imagem é submetida a dilatação e seu resultado é submetido a erosão. Assim, o resultado de uma operação de fechamento em uma imagem normalmente apresenta junção de descontinuidades estreitas, podendo causar o fechamento de pequenas fissuras e preenchimento de lacunas internas dos objetos.

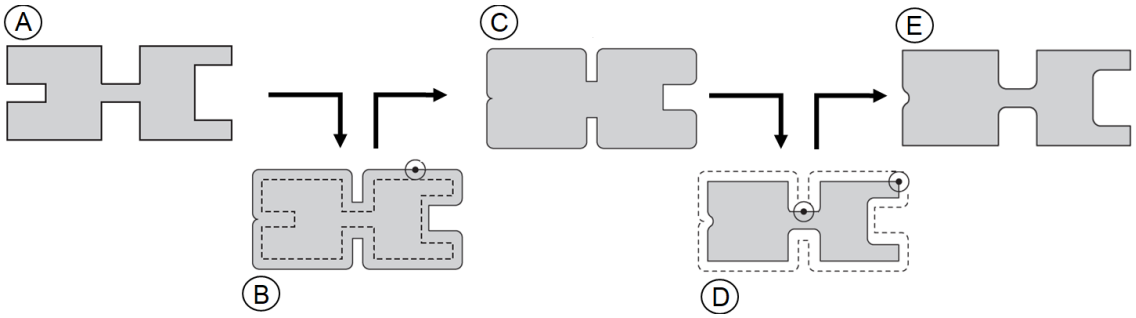


***Figura 21** – Etapas da operação de fechamento de um objeto. A) Objeto original. B) Processo de dilatação, usando um elemento estruturante em forma de disco. C) Resultado obtido após a dilatação. D) Processo de erosão em C, utilizando o mesmo elemento estruturante. E) Objeto após a erosão e representando o resultado final da operação de fechamento.*

5.3.4 MARCAÇÃO DAS IMAGENS ACÚSTICAS

As atividades desta natureza desempenhadas até aqui, contemplam a marcação do conjunto de imagens acústicas disponíveis, indicando o posicionamento dos peixes presentes em cada uma delas. Esse processo possui grande influência tanto na etapa de construção do algoritmo quanto na posterior validação dos resultados obtidos. Isso ocorre pois no desenvolvimento do software a etapa inicial de treinamento, crucial para o bom desempenho da identificação, é necessário utilizar imagens que indiquem corretamente os elementos que devem ser reconhecidos pelo algoritmo. A marcação dos peixes também se torna necessária na fase final de avaliação do desempenho do reconhecimento realizado, uma vez que uma comparação entre o resultado obtido e a marcação em “padrão ouro” deve ser realizada.

A marcação dos peixes nas imagens a serem submetidas para o treinamento da rede neural de identificação possui o intuito de proporcionar maior robustez ao método desenvolvido. Em outras palavras, a marcação prévia das imagens que alimentam a rede neural indica exatamente os elementos de interesse e ressalta as diversas características que devem ser interpretadas. Desta forma, aumenta-se a probabilidade de que somente os peixes sejam identificados e previne que demais objetos e ruídos sejam classificados erroneamente. A marcação prévia dos peixes presentes em cada imagem serve também como um parâmetro para a avaliação do reconhecimento realizado pelo algoritmo, uma vez que se torna a referência para os acertos e erros cometidos.

Atualmente, diversas técnicas podem ser utilizadas para a marcação dos elementos de interesse nas imagens, podendo ser citadas as marcações de formas quadradas ou poligonais planas, cúbicas tridimensionais e a definição de pontos de contorno. O desenvolvimento das marcações dos peixes deste projeto utilizou a técnica de marcações poligonais planas executada a partir da plataforma *ImgLab*. Essa plataforma consiste em um software online onde podem ser realizadas as marcações com posterior armazenamento desses arquivos.

5.3.5 RESULTADOS

Os conceitos teóricos apresentados até o presente momento foram utilizados como base para o desenvolvimento de um algoritmo inicial aplicado à identificação dos peixes. A partir das imagens cruas obtidas pelo sonar, uma conversão para tons de cinza foi realizada e seguida de uma filtragem da imagem. A etapa final consistiu na determinação dos contornos dos objetos, resultando por fim na primeira separação dos peixes com os demais componentes da imagem (Figura 20).

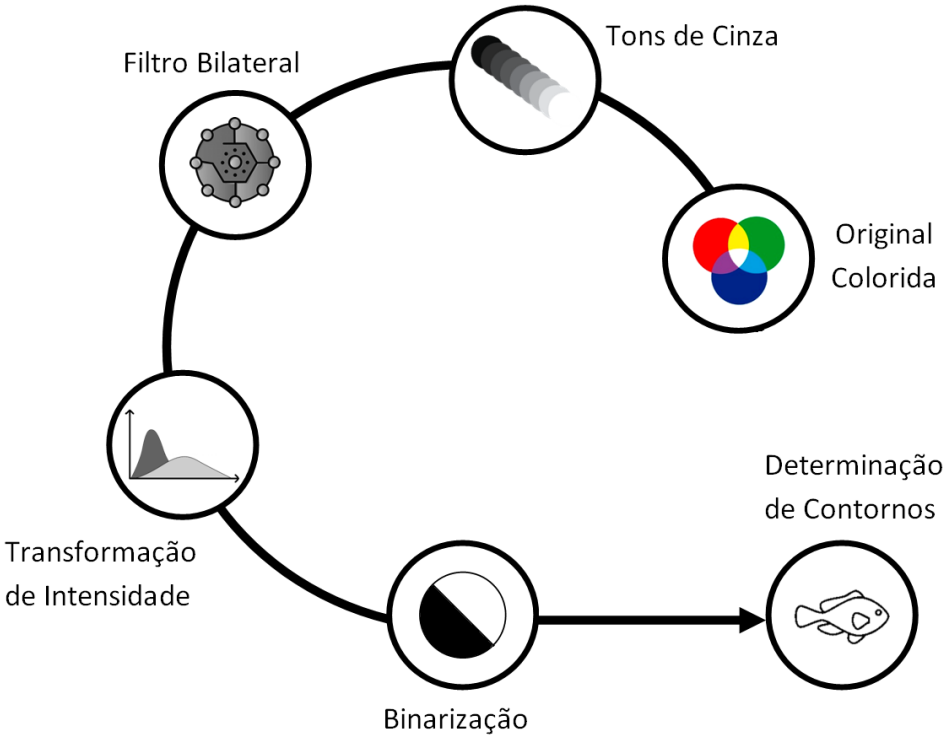


Figura 22 *– Diagrama esquemático das operações realizadas no primeiro algoritmo de identificação de peixes.*

As imagens brutas coloridas obtidas pelo sonar são compostas por diferentes versões da imagem de acordo com a presença de uma determinada cor. Normalmente as imagens são representadas pelo mapa de cores RGB (*red, green, blue*), fazendo com que cada versão da imagem, denominada como canal, indique uma dessas três cores. A combinação entre os canais de cores origina a imagem colorida e no decorrer do processamento todos os canais devem ser manipulados. No caso das imagens do sonar, a informação contida nas cores não apresenta grande relevância para o processo de identificação dos peixes e desta forma optou-se por converter as imagens coloridas para imagens em escala de cinza. A representação em tons de cinza de uma imagem indica que cada *pixel* pode apresentar um valor de intensidade entre 0 e 255, sendo os valores mais próximos de zero indicativos de tons escuros e próximos ao preto enquanto valores próximos a 255 são indicativos de *pixels* brancos. Após a conversão das imagens para escala de cinza, elas foram submetidas à filtragem através do filtro bilateral.

A análise da imagem filtrada e em tons de cinza mostra baixa discriminação dos elementos que a compõem, dificultando inclusive a determinação dos objetos que representam de fato os peixes presentes no canal. A alteração de brilho e contraste de uma imagem são ferramentas muito utilizadas nas etapas de pré-processamento para que, como neste caso, as informações que estão presentes na imagem sejam mais bem definidas. Desta forma, as imagens foram submetidas a uma transformação de potência, utilizando um $\gamma = 1.7$, responsável

por mapear a estreita faixa de valores claros para uma faixa mais ampla. Para fins comparativos, a Figura 22 exemplifica uma das imagens acústicas e seu histograma original, junto a essa mesma imagem após a aplicação da transformação de potência para o alongamento de contraste. O histograma das imagens após o processo de alteração de contraste foi utilizado também para determinar a faixa ideal a ser utilizada para a binarização. As curvas dos histogramas mostram uma maior concentração de *pixels* em intensidades mais baixas e próximas a zero, fato explicado pela imagem ser predominantemente escura. Entretanto, quando os histogramas são analisados conjuntamente, uma proeminência na faixa de intensidade de 60 a 75 é encontrada e indica os pixels que de fato carregam as informações e representam o ruído intrínseco da forma de obtenção de imagens acústicas. Portanto, essa faixa de intensidade foi escolhida para o procedimento de binarização através da função *Canny* da biblioteca OpenCV.

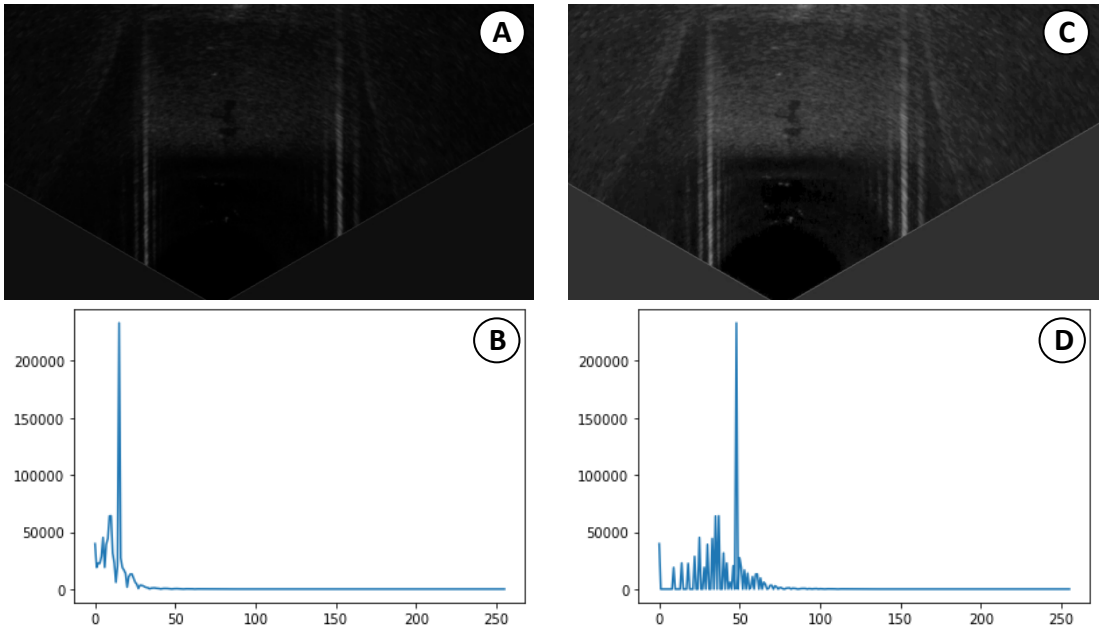


***Figura 23** – As alterações de contraste evidenciam os detalhes existentes nas imagens. A) Imagem acústica original em tons de cinza. B) Histograma da imagem original. C) Imagem originada pela aplicação de transformação de potência com $\gamma = 1.7$. D) Histograma da imagem após o alongamento de contraste.*

A etapa seguinte realizada foi a de identificação das bordas dos objetos das imagens, com foco específico nos elementos que representam os peixes. Para esta tarefa, foi utilizado um comando computacional da biblioteca OpenCV que realiza a identificação dos contornos e também indica caso um objeto seja encontrado dentro das margens de outro, resultado conhecido como hierarquia de contorno. A hierarquia de contorno pode ser útil em processamento a serem realizados futuramente, considerando que os ruídos gerados pelo sonar na criação das imagens podem sobrepor a posição de alguns peixes presentes na imagem. Todos os contornos determinados pela função na imagem foram armazenados, porém, visando reduzir o espaço de memória destinado a isso somente alguns pontos que definem o contorno sem perda de informação foram salvos.

As imagens binarizadas e o resultado da determinação de contornos possuem uma estreita relação, que pode ser analisada de maneira visual através da Figura 23. O processo de binarização é necessário para que a função de determinação de contornos possa ser utilizada, visando que as estruturas da imagem possam ser reconhecidas de maneira otimizada. Entretanto, devido as características de função de binarização aplicada, as imagens obtidas já indicam parcialmente todas as bordas existentes na imagem, ainda que não ocorra distinção entre os elementos que representam os peixes, as paredes do canal ou que são oriundos do ruído da imagem acústica. Desta forma, quando essas imagens são submetidas à função de determinação de contornos o resultado obtido é extremamente semelhante ao encontrado pela

binarização, exceto pelo fato de que as bordas sofrem um espessamento e se tornam mais pronunciadas.

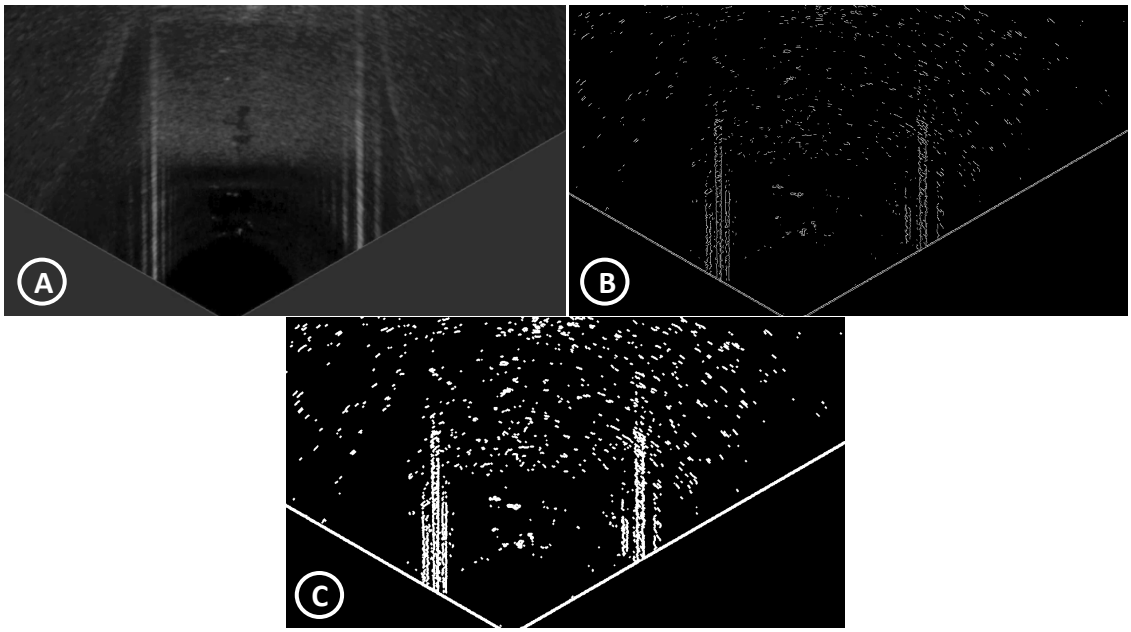


***Figura 24** – Determinação dos contornos existentes na imagem acústica. A) Imagem acústica original em tons de cinza. B) Binarização obtida pela função Canny. C) Contornos obtidos após todos os procedimentos descritos.*

5.3.6 CONCLUSÕES

Os resultados encontrados até o presente momento indicam ao menos dois possíveis caminhos a serem tomados, visando que os ruídos das imagens sejam eliminados e que seja possível determinar todos os peixes presentes no canal independentemente de sua localização. A primeira alternativa consiste no acréscimo de processamentos entre a etapa de binarização e a determinação de contornos. Realizando uma comparação entre a imagem acústica e sua forma binarizada, pode-se notar que as estruturas que representam os peixes permanecem conservadas (Figura 24), bem como são notáveis as estruturas que definem as paredes do canal. Entretanto, o ruído acústico da imagem também se mostra presente, ainda que de forma menos expressiva do que originalmente e em formatos lineares e curtos. O padrão desse ruído, que não define de fato um elemento de contorno que possa ser preenchido, permite que sejam aplicadas operações morfológicas que visam a eliminação somente dessas estruturas e a permanência das formas de interesse.

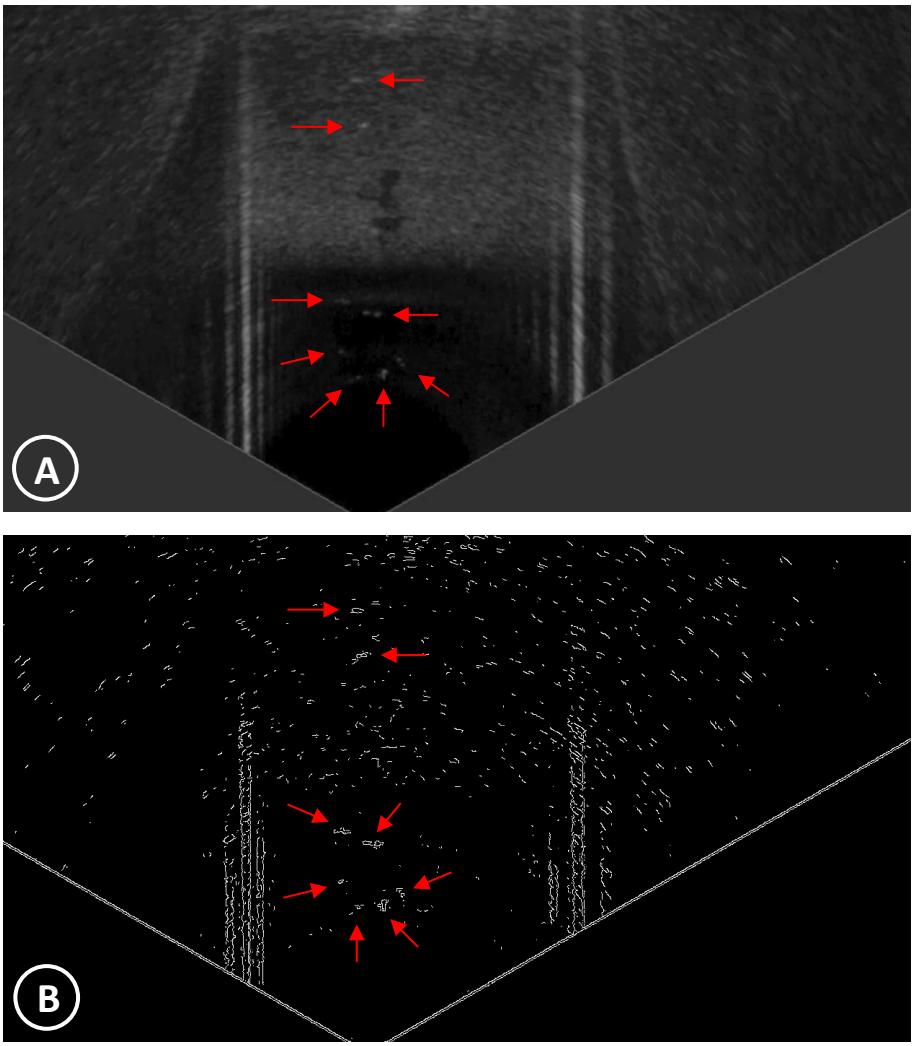


Figura 25 *– Indicação, através das setas vermelhas, das estruturas dos peixes presentes na imagem. A) Imagem em sua versão acústica e com mudança de contraste. B) Imagem binarizada.*

A segunda alternativa, que poderá ser abordada em etapas futuras deste projeto, consiste na fragmentação da imagem em diferentes faixas antes da execução das etapas de pré-processamento. Essa divisão da imagem deve ser realizada com base nos diferentes níveis de ruído, uma vez que esse padrão de distribuição é decorrente do equipamento e forma de aquisição da imagem. Desta forma, as faixas de ruído são invariantes entre as imagens e possuem características próprias (Figura 25), tornando adequada a utilização de diferentes tratamentos de filtragem e alongamento de contraste. A faixa mais inferior das imagens, identificada como nível 1, consiste na região que apresenta o menor índice de ruído, fazendo com que o pré-processamento para a identificação dos peixes seja de menor complexidade. Em contrapartida, a região central da imagem denominada como nível 2 apresenta o maior índice de ruído de forma que pode resultar na ocultação de algumas informações, mais especificamente peixes, ali presentes. Essa faixa da imagem necessita de maior cuidado durante a aplicação dos pré-processamentos, uma vez que o nível de complexidade para a definição dos peixes é maior. A última faixa da imagem, localizada na região superior e denominada como nível 3, apresenta um padrão de ruído intermediário entre os níveis 1 e 2.

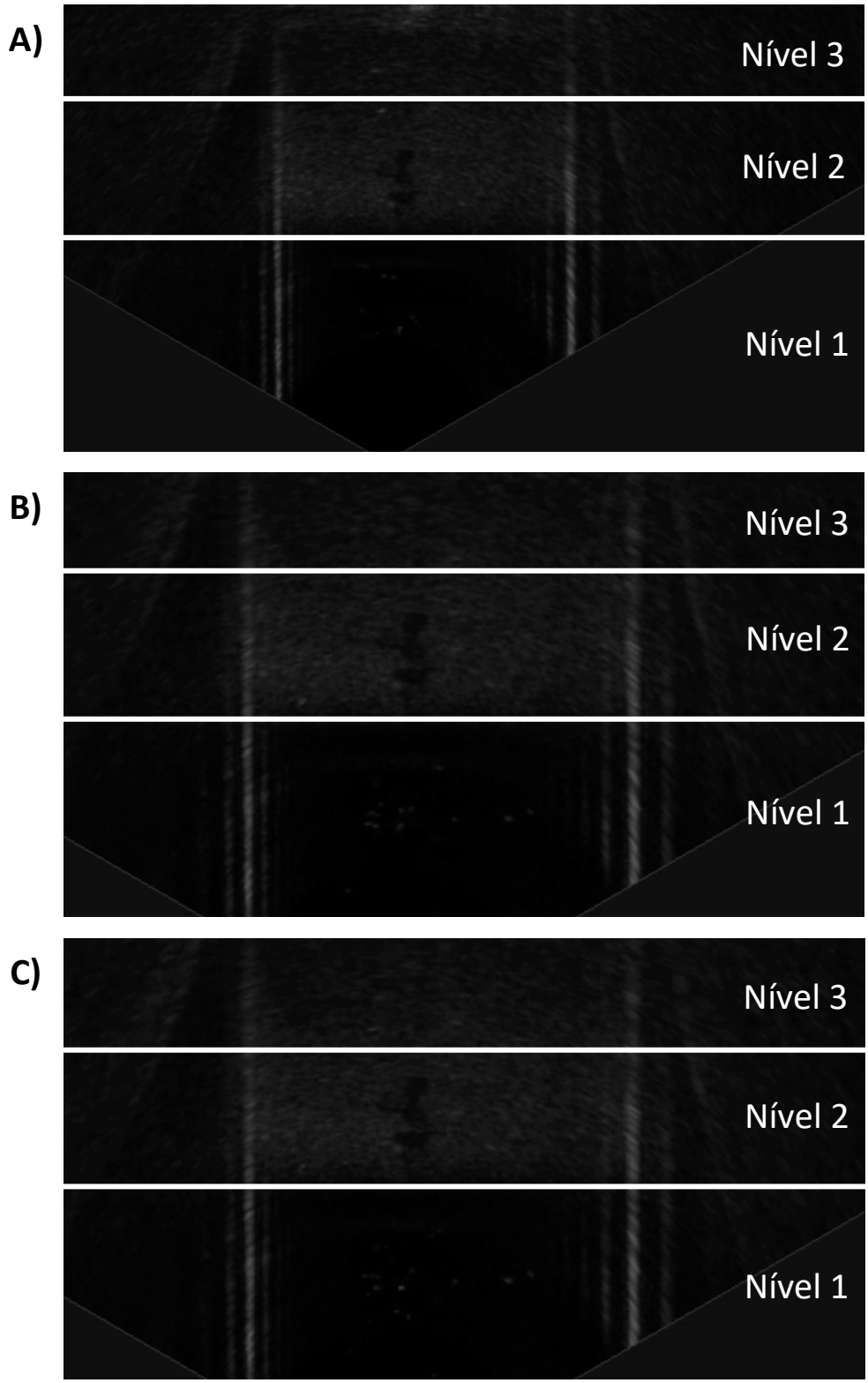


Figura 26 *– Diferentes frames dos registros acústicos, evidenciando a que as faixas de ruído permanecem invariantes. O nível 1 apresenta menor nível de ruído, o nível 3 possui ruído intermediário e o nível 2 alto índice de ruído.*

Em suma, a série de processamentos realizados até o presente momento se mostra eficazes em evidenciar os elementos de interesse nas imagens. Além disso, as reflexões relacionadas ao ruído presente nas imagens e os princípios físicos envolvidos em sua aquisição mostram caminhos promissores para a identificação robusta de peixes em qualquer região da imagem.

5.3.7. MEDIÇÕES

Etapa	Atividade	Custo (R$)
9.7	Aplicação de operadores	40.999,08
9.8	Aplicação de morfologias	48.887,32
	TOTAL	**89.886,40**

Tabela 11 – Recursos Entrega 4

5.3.8. RECURSOS UTILIZADOS

Mês/Ano	Mat. De consumo	Mat. Perm e Equip.	Outros	Serv. 3os	Viagens e diárias	Total Mês
Mar / 20	0,00	0,00	0,00	0,00	0,00	0,00
Abr / 20	0,00	0,00	0,00	0,00	0,00	0,00
Mai / 20	0,00	0,00	0,00	0,00	0,00	0,00

Tabela 12 – Reembolso Entrega 4

5.4. ENTREGA 5

As atividades previstas na entrega (5) que se estabelece, seguem detalhadas:

- Aplicação de classificadores lineares (Correções)

5.4.1 CLASSIFICADOR LINEAR

Desenvolvidas dentro do escopo da Entrega 5 deste projeto estão relacionadas à utilização de algoritmos de aprendizado de máquina nas imagens acústicas. A fase anterior foi composta pela aplicação de métodos de pré-processamento, onde há a manipulação das imagens, visando que os peixes presentes se tornassem evidentes. Os elementos de interesse delimitados previamente foram analisados nessa etapa de forma automática, na busca da identificação de atributos relevantes para a sua caracterização. Esses atributos, que também podem ser denominados como características, consistem em fatores que podem ser determinados nas imagens como por exemplo formato, área de abrangência e textura. A definição dos atributos pode ser realizada de duas formas, sendo uma baseada diretamente na intensidade de cada *pixel* da imagem e a outra através de máscaras específicas. Neste último caso, as máscaras são passadas pela imagem em um processo semelhante ao ocorrido durante

a filtragem e cada uma possui uma disposição particular de valores que permite identificar uma determinada característica.

Cada uma das máscaras, que passam a representar as características de interesse, é aplicada por toda a imagem fazendo com que cada *pixel* seja analisado sob diversas perspectivas. O conjunto das diferentes máscaras dá origem ao que é conhecido como espaço de características e cada pixel da imagem deve ser alocado nesse espaço, sendo representado como uma combinação dessas diferentes características. De maneira geral, os *pixels* que constituem os peixes tendem a apresentar atributos semelhantes e se agruparem em uma determinada região do espaço de características (Figura 26). Assim, pode-se utilizar um algoritmo que determine uma forma de separar esses conjuntos, denominados como classes, e que representa de fato a identificação dos peixes na imagem. O *Support Vector Machine* (SVM) é um dos algoritmos mais utilizados para a construção de um hiperplano de separação das classes no espaço de características. A preferência na utilização do SVM se dá por ele buscar a otimização desse hiperplano, uma vez que é calculada a distância entre a superfície e cada classe visando que essa distância se mantenha equilibrada em ambos os lados. Além disso, no processo de definição do hiperplano os atributos que mais influenciam na classificação são determinados e assim os demais podem ser desconsiderados em etapas futuras.

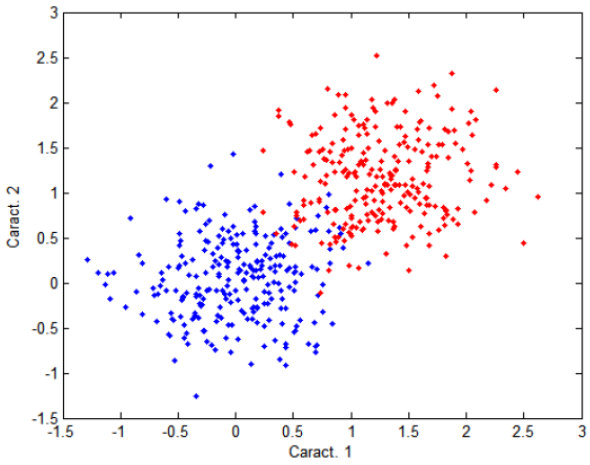


***Figura 27** – Representação ilustrativa de um espaço de características composto por somente duas características. Cada elemento azul ou vermelho indica a combinação dos atributos para um pixel da imagem. Os pontos vermelhos podem indicar o grupo de pixels que formam os peixes, enquanto os azuis são os demais pixels da imagem, sendo visível a separação entre esses conjuntos.*

Levando em consideração os aspectos fundamentais do SVM, que são a identificação otimizada do hiperplano e a definição das características que mais contribuem para a classificação, foi utilizada nessa etapa do projeto a função *Cascade Classifier* inclusa na biblioteca *OpenCV*. Nesta função, diversas máscaras podem ser utilizadas para investigar os atributos de interesse, como o conjunto de máscaras conhecidas como características *Haar* (Figura 27), e dão origem ao espaço de características. Diferentemente do algoritmo original do SVM, o *Cascade Classifier* separa os atributos em diferentes níveis de forma que caso ocorra uma falha de compatibilidade da região em um determinado nível, essa região não é submetida ao próximo nível de atributos. Esse processo é aplicado em todas as imagens do conjunto de treinamento do classificador e a cada ciclo os pesos de cada atributo são atualizados de forma a criar um aprendizado do padrão das imagens. Desta forma, as principais máscaras que contribuem para a classificação dos peixes podem ser identificadas e utilizadas posteriormente.

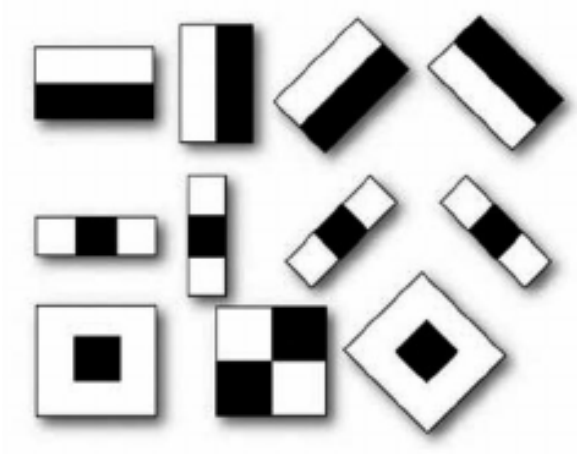


***Figura 28** – Máscaras que compõem o conjunto de características Haar (Haar Features).*

Em suma, a série de processamentos realizados até o presente momento se mostra eficazes em evidenciar os elementos de interesse nas imagens. Além disso, as reflexões relacionadas ao ruído presente nas imagens e os princípios físicos envolvidos em sua aquisição mostram caminhos promissores para a identificação robusta de peixes em qualquer região da imagem.

5.4.2. ALGORITMOS DE RASTREAMENTO

A identificação dos peixes presentes no tubo de sucção representa o principal objetivo de desenvolvimento deste projeto, uma vez que desta forma torna-se possível sua quantificação de maneira automatizada. Entretanto, o puro reconhecimento dos animais nas imagens pode causar ambiguidade nos dados obtidos, considerando que sem o devido monitoramento o mesmo peixe pode ser contabilizado por mais de uma vez em um mesmo registro acústico. Os arquivos utilizados para o desenvolvimento do software, bem como em sua futura aplicação em tempo real, consistem em vídeos que essencialmente podem ser definidos como um conjunto de imagens obtidas sequencialmente. Cada uma das imagens que compõem um vídeo é denominada como *frame*, e representa um instante específico de todo o registro temporal realizado. Os *frames* devem passar por todas as etapas de processamento individualmente, incluindo o processo de quantização dos peixes, de forma que se torna necessário um método de rastreamento dos animais no decorrer do vídeo.

Os algoritmos de rastreamento, também comumente chamados de algoritmos de *tracking*, buscam identificar através de um conjunto de imagens elementos que estão em movimento, possibilitando a identificação da posição, velocidade e trajetória realizada. No contexto dos registros realizados pelo sonar, os peixes que adentam o canal encontram-se em constante movimentação, de forma que o a utilização desse tipo de algoritmo se torna razoável. Entretanto, para que o rastreamento da trajetória do objeto seja efetivo é importante que os elementos de interesse sejam previamente identificados e isolados. As técnicas mais utilizadas para essa etapa de detecção do objeto na imagem são a segmentação, através do pré-processamento, a remoção de fundo ou também a aplicação de *template matching*. A detecção dos elementos dinâmicos através da remoção de fundo consiste em identificar a imagem de fundo, característica de quando os elementos de interesse não se encontram em cena, e obter a diferença com cada quadro do vídeo de forma que sejam mantidos apenas os objetos que

encontram-se em movimento. Em contrapartida, a utilização de *template matching* utiliza um padrão do objeto como referência, varrendo as imagens em busca de regiões semelhantes e que devem ser demarcadas para o rastreamento.

O acompanhamento da trajetória dos animais possibilita uma contagem robusta, uma vez que a entrada ou saída de um peixe do canal pode ser facilmente verificada. A atualização da quantidade de peixes, portanto, passa a ser realizada somente quando um desses eventos ocorre, ainda que os animais sejam analisados a cada frame. A literatura apresenta diversas alternativas de algoritmos de *tracking,* porém, a seguir, serão apresentados os métodos mais utilizados que são o filtro Kalman, fluxo óptico, *mean-shift* e *CAMShift tracking*.

5.4.2.1. Filtro Kalman

O filtro Kalman consiste em uma técnica de rastreamento de objetos que se baseia na realização de uma série de estimativas. A premissa fundamental deste algoritmo é de que a sequência de imagens com elementos em movimento trata-se de um sistema dinâmico linear, que pode ser descrito essencialmente através de diversos estados e suas transições (KALMAN, 1960). Desta forma, no contexto dos *frames* do vídeo, o estado consiste em um conjunto de variáveis e equações a serem observadas, como por exemplo um modelo para a posição e velocidade dos peixes, com o intuito de realizar estimativas de condições futuras.

A análise realizada através do filtro Kalman pode ser subdividida em duas etapas, sendo a primeira de predição e a segunda de correção. Na etapa de predição, a condição das variáveis de interesse que compõem o estado é estabelecida e por conta disso é comum definir essa etapa como estado *a priori*. Esse estado inicial consiste no *frame* corrente, a partir do qual se inicia a análise. Em seguida, devido a transição das imagens uma medida é realizada e uma nova estimativa já corrigida é obtida, denominada como estado *a posteriori* (FEIJÓ, 2016). A condição final do estado *a posteriori* é usada como estado *a priori* para que um novo ciclo de estimativas seja iniciado, de forma que essa sequência de análises seja repetida através de todos os frames do vídeo.

5.4.2.2. Fluxo Óptico

O fluxo óptico pode ser definido como a avaliação do movimento de objetos entre *frames* consecutivos de um vídeo. A análise comparativa leva em consideração tanto o frame corrente quanto o frame imediatamente anterior, uma vez que se busca identificar alguns pontos específico em ambas as imagens, denominados como “cantos”. A detecção dos cantos consiste em identificar os *pixels* da imagem que apresentam grande variação de intensidade em duas direções, ou seja, o *pixel* representa uma região de borda com mudança de intensidade tanto na direção *x* quanto na direção *y*. Uma vez que esses *pixels* de canto são identificados, o monitoramento do movimento é realizado com base em seu posicionamento, de forma a gerar um mapeamento vetorial do deslocamento. Isso ocorre pois quando analisamos dois instantes de tempo consecutivos temos o produto entre o gradiente da imagem e a velocidade que deve ser equivalente a variação de intensidade entre os *frames* (Nanonets, 2019).

A medida do fluxo óptico pode ser realizada através de técnicas diferenciais, técnicas de correlação e técnicas baseadas em frequência, sendo o primeiro tipo de técnica o mais utilizado (MARTINS, 2020). Os algoritmos baseados em técnicas diferenciais partem da premissa que *frames* diferentes pertencentes a mesma sequência possuem intensidades aproximadamente

constantes, uma vez que o intervalo de tempo será pequeno e o deslocamento mínimo. Um dos principais algoritmos desse tipo de técnica é o Algoritmo Lucas-Kanade (ou Algoritmo LK), que além da premissa inicial apresentada, também assume que pontos pertencentes a mesma superfície possuem movimento semelhante e que no decorrer dos *frames* não haverá mudança de brilho, contraste ou cor nas imagens.

5.4.2.3. Mean-shift Tracking e CAMShift Tracking

Os métodos de rastreamento *mean-shift* e *continuous adaptive mean-shift* (*CAMShift*) se baseiam na identificação de regiões na imagem que possuem uma certa distribuição de intensidades dos *pixels*. Esse padrão de distribuição de intensidades é obtido através de um janelamento inicial, centralizado no objeto que se deseja acompanhar o movimento. Assim, com base nas características de intensidade da região de interesse (ROI), a cada novo frame o algoritmo realiza uma busca interativa que procura identificar a área que mais se assemelha à distribuição de forma a rastear a nova posição do objeto. O método *mean-shift* mantém fixo o tamanho da janela analisada, sendo essa característica também um de seus limitantes. As mudanças de escala do objeto, devido a aproximação ou afastamento da câmera de aquisição das imagens, podem causar alterações significativas na distribuição de intensidades de forma que o rastreamento através da janela fixa é comprometido.

Visando minimizar o impacto do janelamento fixo, o método *CAMShift* foi desenvolvido, sendo uma variação do método *mean-shift*. Neste caso, o tamanho da janela utilizada para a definição da distribuição de intensidades a ser buscada na imagem é ajustado a cada novo frame. Desta forma, mesmo que uma mudança de escala e distribuição de intensidades ocorra, a mudança constante do tamanho da janela de comparação realiza uma compensação desses efeitos. Esses algoritmos que se baseiam na distribuição de intensidades são boas alternativas para o rastreamento de objetos simples e de aparência constante, ainda que a necessidade de seleção da janela por parte do usuário possa representar uma redução da robustez do rastreamento (EMAMI & FATHY, 2011)

5.4.3. SISTEMA SONAR

O sistema SONAR utilizado para o registro da atividade aquática no interior do tubo de sucção possui como base o princípio da ecossondagem. Neste tipo de técnica, o equipamento realiza a emissão de ondas de ultrassom, que interagem com os elementos do meio e refletem de forma a serem captadas novamente. O sinal obtido pelo retorno da onda tem uma intensidade diferente da emitida, bem como uma alteração em sua frequência, sendo esses os principais parâmetros responsáveis pela formação das imagens acústicas. A emissão e recepção das ondas acústicas ocorre a partir de transdutores do tipo piezelétrico, que realizam a conversão entre energia elétrica e mecânica. As ondas mecânicas geradas pelo transdutor são emitidas na forma de feixes e se propagam através de frentes de ondas esféricas, que tendem a se dissipar igualmente através do meio (Figura 28).

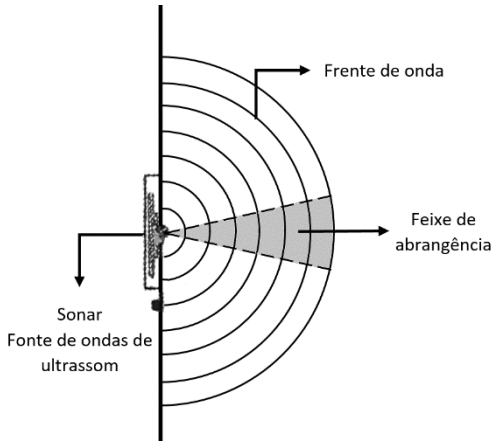


Figura 29 – *Propagação de frentes de onda de ultrassom a partir do feixe emitido. Fonte: Elaborado pelos autores.*

Devido à disposição dos transdutores no equipamento, ainda que as ondas se propaguem em todas as direções a partir de sua origem, a capacidade de obter informações da região se limita a área de abrangência do feixe. Isso ocorre pois mesmo que as frentes de onda encontrem regiões que excedem o alcance do feixe, a onda de retorno pode não chegar até o transdutor do equipamento ou ter uma amplitude muito reduzida a ponto de ser irrelevante para a construção da imagem.

5.4.4. RESULTADOS

O entendimento do processo de formação das imagens acústicas é essencial para que informações de grande relevância para o monitoramento dos peixes não sejam perdidas. O posicionamento do sonar multifeixes com relação a estrutura do tubo de sucção, por exemplo, é decisivo para a correta identificação da ictiofauna. De forma geral, o tubo de sucção realiza a ligação entre as turbinas de geração de energia e o canal de fuga, de forma que a água utilizada possa retornar ao rio (Figura 29). Devido à aglomeração de peixes nas proximidades da barragem, principalmente no canal de fuga, os animais podem adentrar no tubo de sucção durante algumas manobras de funcionamento da usina, correndo assim grande risco (GODINHO & LOURES, 2016.).

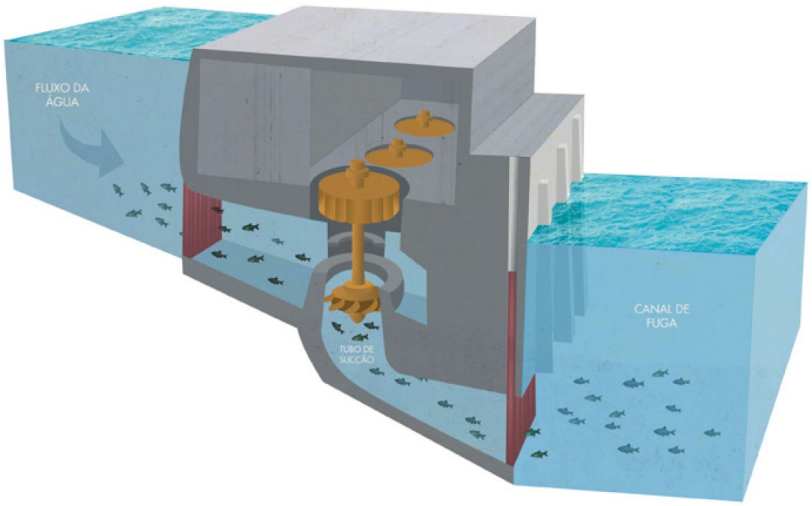


Figura 30 – *Estrutura de uma casa de força de hidrelétrica, onde o movimento das turbinas direciona o fluxo de água para o tubo de sucção que desemboca no canal de fuga. Fonte: GODINHO & LOURES, 2016.*

O movimento adotado pelos peixes possui caráter contracorrente, levando-os do canal de fuga em direção às turbinas. Portanto, as imagens acústicas obtidas através do sonar devem cobrir a maior extensão possível do tubo de sucção, sendo o interesse maior nas regiões mais profundas do tubo que se encontram próximas das turbinas. Tendo isso em mente, a Figura 30a apresenta o posicionamento ideal do sonar, onde o feixe de ondas de ultrassom é emitido de forma diagonal em direção a região das turbinas, formando assim um ângulo agudo com relação a parede de fixação do equipamento. Este direcionamento do feixe permite toda a abrangência necessária do tubo de sucção, em contraste com o resultado que seria obtido caso as ondas de ultrassom fossem liberadas de forma perpendicular ao equipamento onde somente uma secção transversal do tubo seria registrada (Figura 30b).

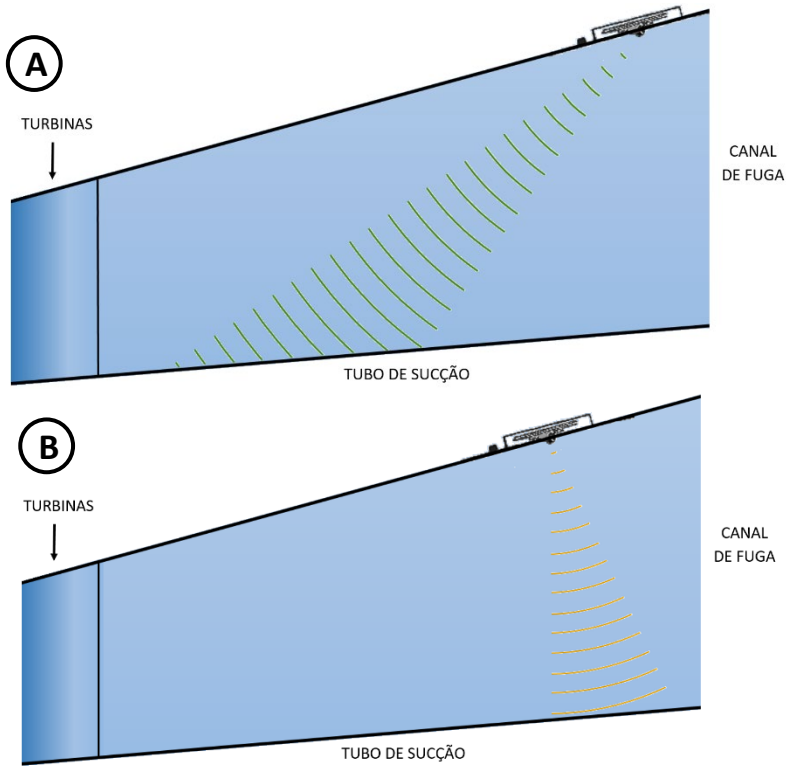


Figura 31 – *Comparação das áreas de abrangência dos feixes de ultrassom para aquisição das imagens. A) Emissão angulada, permitindo o registro de toda a extensão do tubo de sucção. B) Emissão perpendicular, possibilitando a análise somente de uma secção transversal do tubo. Fonte: Elaborado pelos autores.*

A área englobada pelo feixe emitido pelo sonar pode ser subdividida em quatro diferentes zonas de acordo com a estrutura física do tubo de sucção e seu nível de profundidade com relação ao canal de fuga (Figura 31). Essa divisão realizada de acordo com a incidência das ondas de ultrassom pode ser usada para a identificação da área correspondente nas imagens acústicas. A Zona 1, evidenciada na Figura 32, representa a região mais próxima ao canal de fuga, indicando o início do feixe de ultrassom e sendo localizada na área inferior das imagens. As Zonas 2 e 3 podem ser encontradas na região central dos frames e representam diferentes profundidades entre si, sendo áreas consecutivas no tubo de sucção. A porção final do feixe, denominada como Zona 4, engloba uma área maior em decorrência das características de espalhamento do sinal. A região, que é delimitada na porção superior das imagens do sonar, corresponde a parte inicial do tubo de sucção que é mais próxima as turbinas, sendo, portanto, a área mais crítica para a identificação dos peixes.

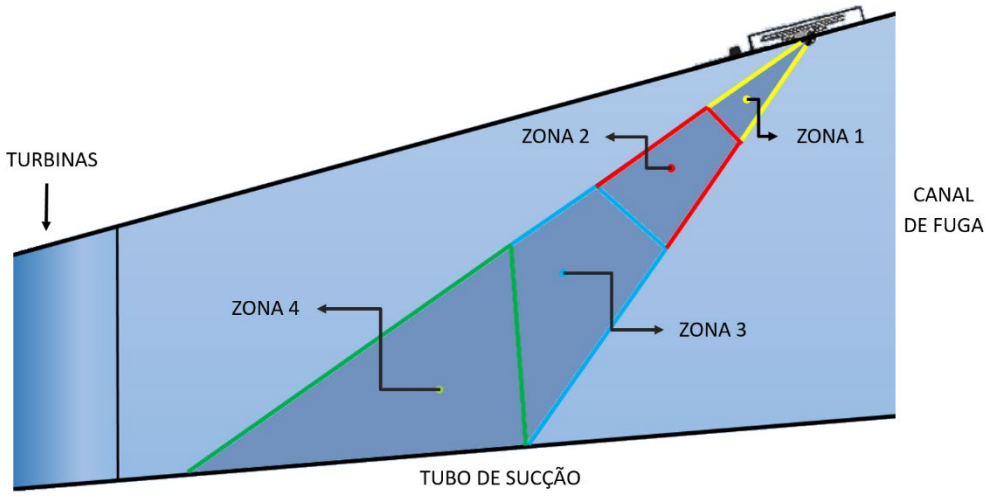


Figura 32 – *Subdivisão da região de abrangência do sonar em 4 diferentes zonas de acordo as características do tubo de sucção. Fonte: Elaborado pelos autores.*

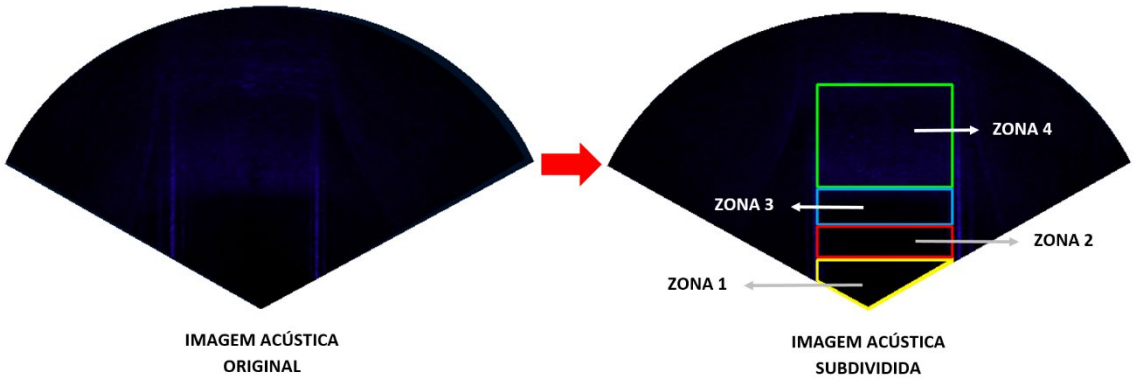


Figura 33 – *Identificação das diferentes zonas nos registros acústicos, visando a realização de cortes na imagem. Fonte: Elaborado pelos autores.*

A definição das diferentes regiões do tubo de sucção nas imagens obtidas pelo sonar facilita o processo de monitoramento dos peixes, uma vez que permite que as etapas de processamento computacional sejam adequadas e otimizadas. As imagens acústicas apresentam diferentes níveis de ruído em sua extensão e, sendo assim, a fragmentação da imagem em quatro diferentes zonas possibilita que tratamentos distintos sejam aplicados de forma a evidenciar de forma aprimorada os peixes. A separação das zonas presentes nas imagens foi realizada de forma automatizada em cada frame que compõe o registro do sonar, consistindo em um corte da região de interesse para que as demais etapas de pré-processamento fossem aplicadas. As coordenadas que definem cada uma das zonas foram determinadas de maneira empírica, através da análise da trajetória do feixe emitido e em conjunto com a equipe técnica. As imagens acústicas produzidas pelo sonar possuem grande dimensão, porém, após o corte referentes a cada zona as imagens produzidas apresentam uma redução drástica na quantidade de pixels e por isso um redimensionamento foi aplicado. O processo de modificação da escala espacial de cada corte manteve as proporções originais através da aplicação de uma interpolação cúbica, visando que distorções drásticas não afetassem as imagens.

O isolamento de cada uma das quatro zonas da imagem acústica consiste no ponto de partida para as demais etapas da metodologia apresentada no Capítulo 5.3.5, agora aplicadas em cada região separadamente (Figura 33). Imediatamente após o corte das imagens as mesmas são convertidas para apresentarem somente tons de cinza, sendo sucedidas pela

aplicação do filtro bilateral, alteração de brilho e contraste, binarização e, por fim, a determinação dos contornos que definem os peixes. A etapa de transformação de intensidades é a mais importante para essa nova metodologia de definição de contornos nas diferentes zonas, pois uma vez que cada região apresenta níveis de ruído característicos, os tratamentos adotados devem ser distintos. Anteriormente, utilizando os frames completos para a definição dos contornos, era evidente a necessidade de adequações na metodologia nas diferentes faixas de ruído da imagem uma vez que a definição de contornos possuía alta acurácia em algumas regiões e baixa capacidade em outras.

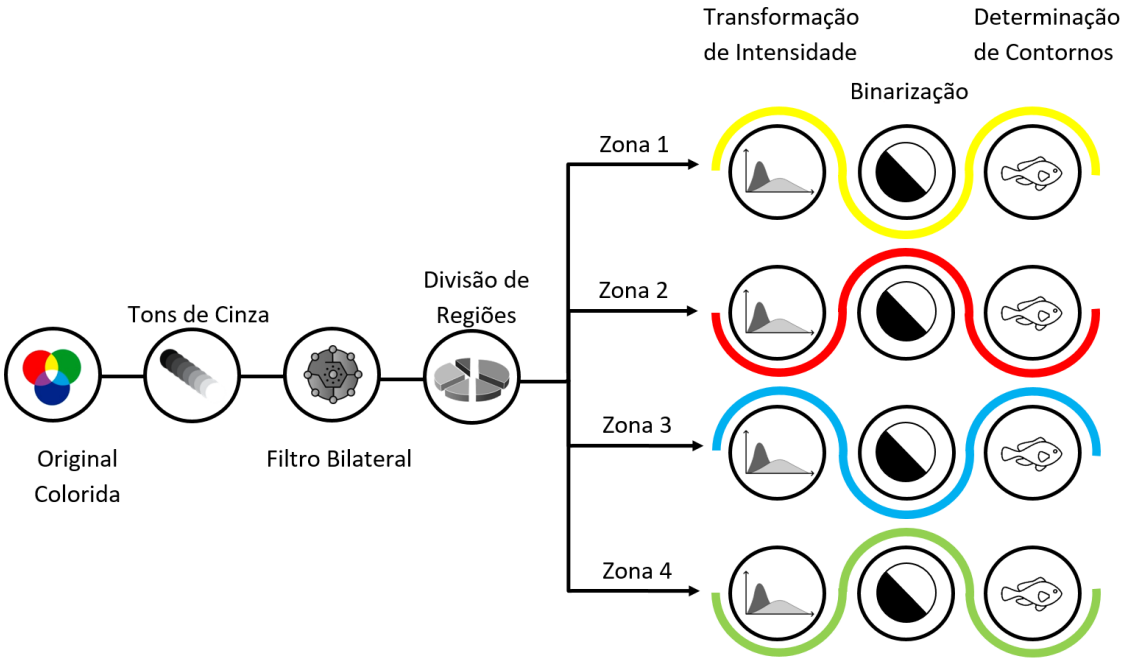


Figura 34 – *Diagrama esquemático das operações realizadas nas imagens acústicas, incluindo a divisão em quatro áreas distintas para a aplicação dos pré-processamentos de imagem. Fonte: Elaborado pelos autores.*

A divisão da imagem acústica em zonas de acordo com a posição no tubo de sucção favorece também a utilização de algoritmos de tracking, usados para o monitoramento dos peixes. O rastreamento dos animais, visando robustez em sua quantificação, deve ser realizado em cada um dos cortes obtidos através dos frames, mantendo em mente que não há uma única direção de movimento. Os peixes vindos do canal de fuga tendem a adentrar no tubo de sucção e se direcionar às turbinas, porém devido sua liberdade diversas trajetórias podem ser observadas. O movimento esperado dos animais nas imagens acústicas é de transição entre as zonas definidas anteriormente, sendo de entrada ou saída, porém é necessário lembrar que o feixe do sonar não abrange toda a área do tubo. Desta forma, a transição dos peixes pode ocorrer também em cada zona delimitada da imagem acústica, uma vez que áreas não registradas pelo equipamento também pertencem ao tubo e podem apresentar peixes (Figuras 34 e 35).

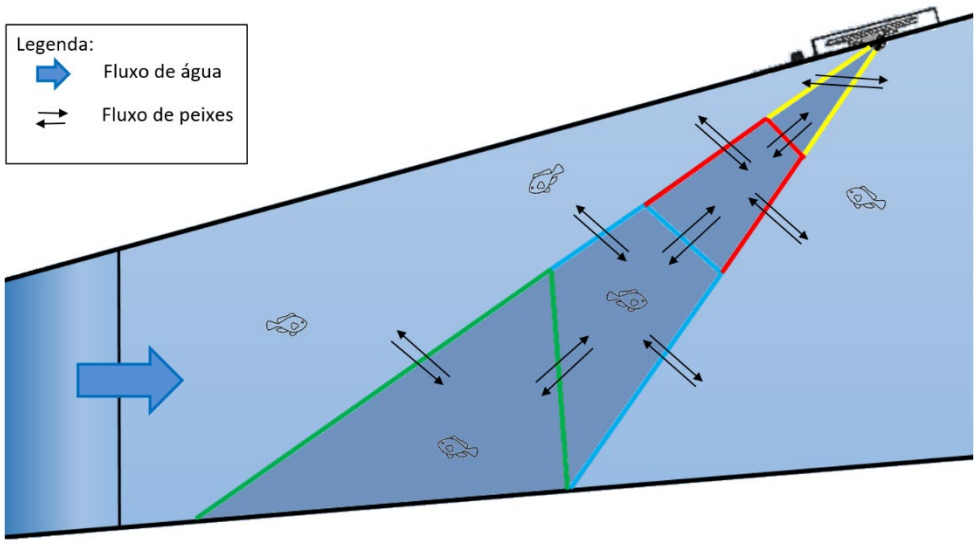


Figura 35 – *O movimento realizado pelos peixes no interior do tubo de sucção pode traçar diversas trajetórias distintas, permitindo que em os animais entrem e saiam da região englobada pelo feixe do sonar em qualquer zona e também transitem entre as zonas. Fonte: Elaborado pelos autores.*

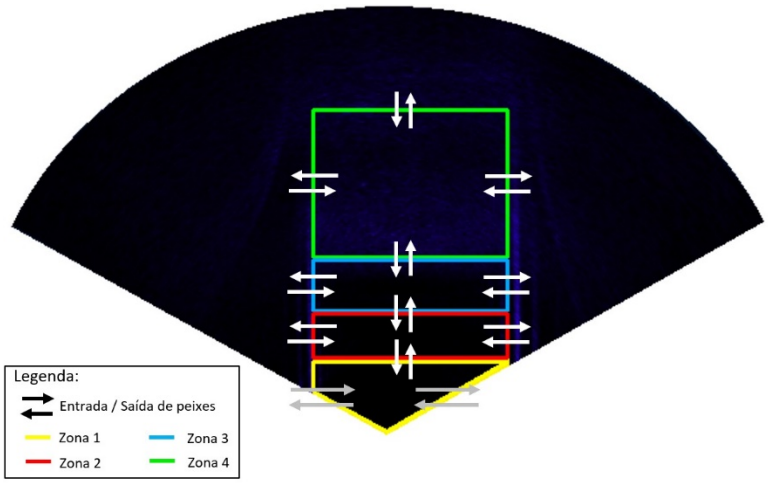


Figura 36 – *As quatro zonas de separação nas imagens acústicas permitem que os peixes entrem ou saiam de cada região tanto pelas laterais quanto pelas fronteiras inferior e superior, sendo este fato devido a presença de animais em áreas não mapeadas pelo sonar. Fonte: Elaborado pelos autores.*

5.4.5. CONCLUSÕES

A análise aprofundada do processo de formação das imagens acústica junto a identificação da estrutura do tubo de sucção permitiu uma subdivisão adequada das imagens acústicas em diferentes zonas. Devido a forma de emissão angulada das ondas de ultrassom, todos os registros apresentam o mesmo padrão de distribuição onde a região superior da imagem representa áreas mais distantes do sonar, sendo, portanto, mais profundas no tubo de sucção. Enquanto isso, a parte inferior da imagem indica regiões anexas ao equipamento e próximas ao canal de fuga, de onde os peixes se originam.

Ainda que os peixes adentrem ao tubo de sucção e se movimentem, predominantemente, em direção as turbinas, a liberdade dos animais permite que o tubo seja explorado em todas as direções com diversas trajetórias. Assim, a identificação dos peixes pode

ser realizada tanto nas regiões de transição entre zonas como nos limites laterais, uma vez que esses animais podem ser encontrados em áreas do tubo que não são abrangidas pelo feixe emitido pelo sonar.

Ademais, a fragmentação dos registros acústicos viabiliza que as etapas de pré-processamento sejam otimizadas para as características de cada zona, como por exemplo o nível de ruído. A variação dos tratamentos aplicados permite que a identificação dos peixes possa ser realizada com maior acurácia, influenciando também nos resultados obtidos após a aplicação do algoritmo de tracking. O reconhecimento da posição dos peixes no tubo de sucção através das diferentes zonas permite também que os animais em maior risco sejam quantificados, de forma que estratégias de resgate possam ser planejadas.

5.4.6. MEDIÇÕES

Etapa	Atividade	Custo (R$)
9.9	Correções	38.409,16
	TOTAL	**38.409,16**

Tabela 13 – Recursos Entrega 5

5.4.7. RECURSOS UTILIZADOS

Mês/Ano	Mat. De consumo	Mat. Perm e Equip.	Outros	Serv. 3os	Viagens e diárias	Total Mês
Jun / 20	0,00	0,00	0,00	0,00	0,00	0,00
Jul / 20	0,00	0,00	0,00	0,00	0,00	0,00
Ago / 20	0,00	0,00	0,00	0,00	0,00	0,00

Tabela 14 – Reembolsos Entrega 5

5.5. ENTREGA 6

5.5.1. MEAN-SHIFT E CAMSHIFT

Os algoritmos *Mean-Shift* e *Continuous Adaptative Mean-Shift (CAMShift)* consistem em métodos de rastreamento, ou *tracking*, que se baseiam em uma análise não-paramétrica das características que definem a imagem. O algoritmo *Mean-Shift* foi proposto em 1975 por Fukunage e Hostetler com o intuito de utilizar estimadores não-paramétricos de densidade de

probabilidade para identificar *clusters* presentes no espaço de características corrente. A característica mais comum de ser utilizada é o histograma da imagem, ainda que outros parâmetros possam ser utilizados, e deve ser representado como uma distribuição de probabilidade (BRADSKI, 1998). Assim, através da utilização de *kernels* específicos com tamanhos de janelas previamente determinadas, o *mean-shift* realiza a estimativa da densidade de probabilidade dando origem a um vetor que indica a direção de crescimento do gradiente (YAMASAKI, 2019; CHEN, 2018; SUN, 2012).

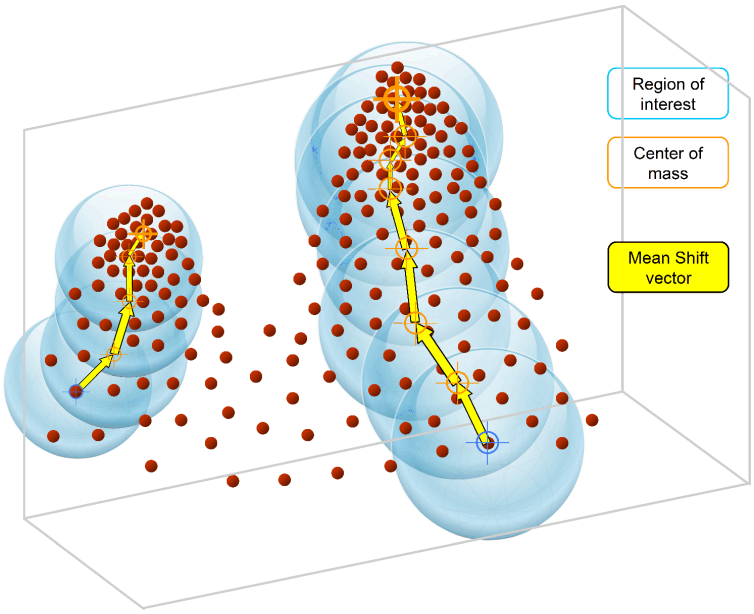


Figura 36 *– Processo realizado pelo algoritmo mean-shift, direcionando a janela de abrangência para as regiões com maior densidade de probabilidade, sendo a direção do gradiente representado pelas setas amarelas. Fonte: CHEN (2018).*

A estimativa de densidade tem como resultado um vetor de deslocamento que pode ser determinado através da Equação 1, onde *x* indica o ponto de partida, x_j os demais *n* pontos considerados no processo, *h* representa a dimensão da janela e a função *g* consiste no *kernel* utilizado. A compreensão dessa estimativa se torna mais simples quando consideramos as etapas de processamento realizadas pelo algoritmo *mean-shift*. Inicialmente, deve-se selecionar uma janela que englobe todo o objeto alvo a ser rastreado pelo algoritmo, considerando que as características de textura e histograma nessa região serão adotadas como "padrão ouro" para os demais *frames* do vídeo. Assim, uma vez que o tamanho da região e as características do alvo já foram extraídos no primeiro *frame*, bem como sua posição identificada, a próxima etapa consiste em utilizar o *kernel* para percorrer toda a extensão do *frame* e obter os atributos necessários. O conjunto desses atributos dá origem a uma dispersão de pontos no espaço de características, como o representado pelas esferas vermelhas da Figura 36, com as quais o cálculo da Equação 1 pode ser realizado. Desta forma, pode-se determinar a direção $m_{h,G}(x)$ média e assim definir qual será a posição adotada como ponto de partida para comparação com o próximo *frame*. O procedimento descrito, e evidenciado pela Figura 37, é executado em todos os frames que constituem o vídeo, de forma que assim o objeto alvo possa ser rastreado e acompanhado integralmente.

$$m_{h,G}(x) = \frac{\sum_{j=1}^{n} x_j g\left(\left\|\frac{x - x_j}{h}\right\|^2\right)}{\sum_{j=1}^{n} g\left(\left\|\frac{x - x_j}{h}\right\|^2\right)} - x \qquad \text{(Eq. 1)}$$

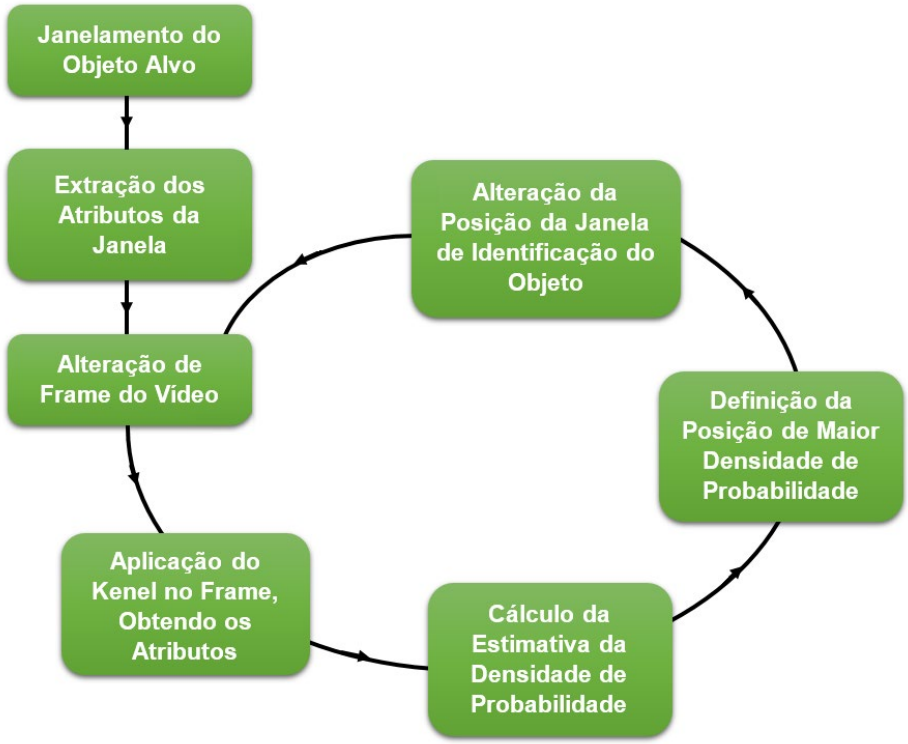


Figura 37 *– Diagrama esquemático do algoritmo mean-shift. Fonte: Elaborado pelos autores com base nas informações de SUN (2012) e BRADSKI (1998).*

A execução prática do algoritmo *mean-shift* pode ser observada através da Figura 38, extraída do trabalho desenvolvido por Sun (2012). O conjunto das imagens mostra o objeto alvo, bem como seu rastreamento em diversos *frames* indicando tanto a posição inicial quanto a final a partir de retângulos. As regiões delimitadas pelas caixas indicam a janela anterior adotada como referência e a janela atual encontrada, que será utilizada como referência para o próximo *frame*. Devido as características intrínsecas do algoritmo, evidenciadas pelo experimento mostrado, o tamanho da janela de referência se mantém inalterado durante todo o processamento. Entretanto, o movimento do objeto na sequência de *frames* pode causar sua oclusão, mudanças de escala ou alterações de brilho e contraste no caso de imagens coloridas, que interferem diretamente no padrão do histograma obtido como atributo (LIU, 2018; BRADSKI, 1998). Assim, levando esses aspectos em consideração Bradski (1998) propôs uma alteração no algoritmo *mean-shift* original, conhecido como *continuous adaptative mean-shift (CAMShift)*. Como é sugerido pelo nome, esse novo método sugere que o tamanho da janela e o atributo de referência sofram modificações dinâmicas bem como as ocorridas no objeto alvo no decorrer do vídeo. Desta forma, a principal diferença entre os algoritmos está na etapa de definição da janela de referência utilizada para o rastreamento *frame* seguinte, uma vez que no *mean-shift* o tamanho se mantém fixo enquanto no *CAMShift* há um ajuste relacionado diretamente com o padrão de distribuição atual dos atributos.

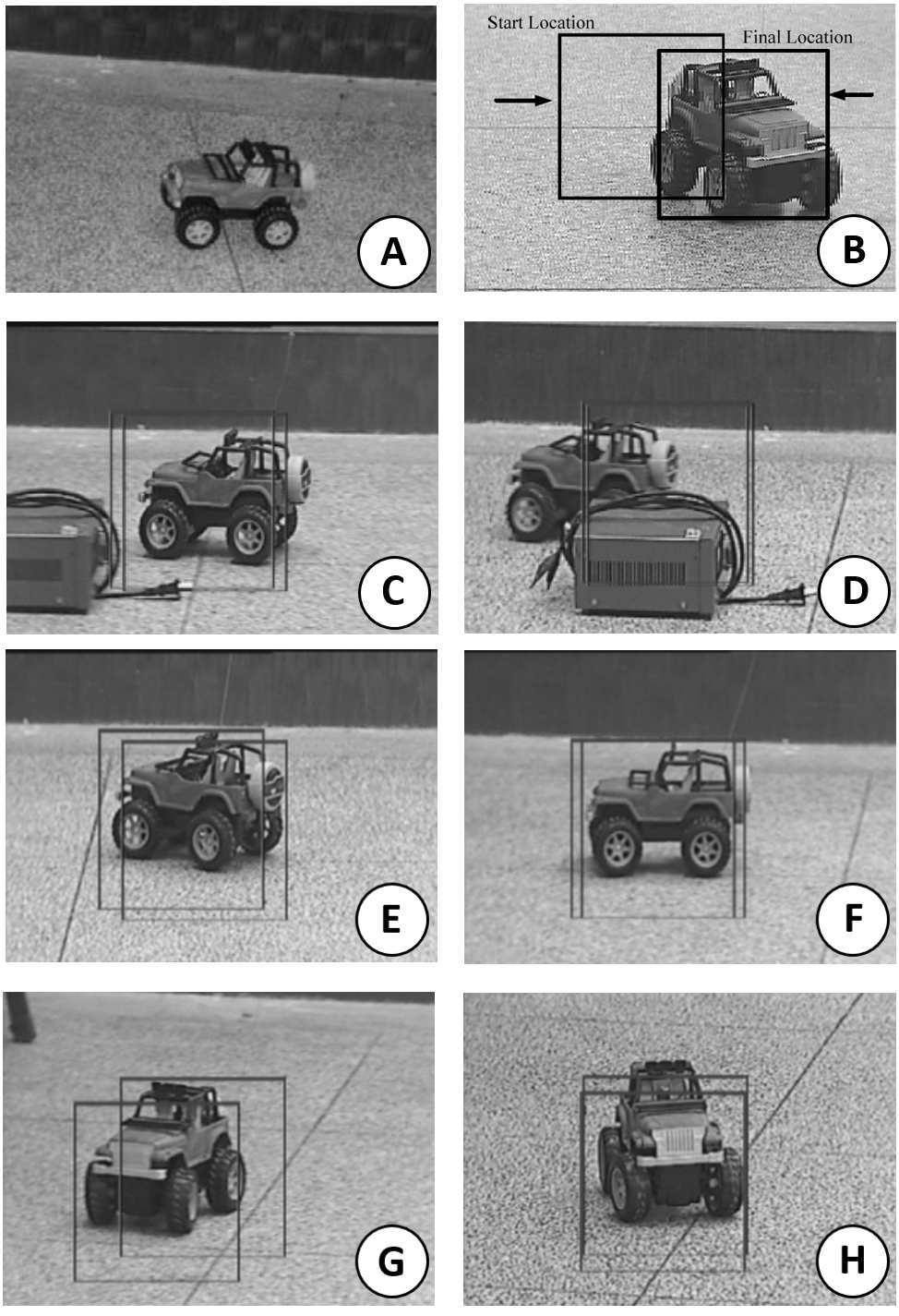

A
Start Location
Final Location
B
C
D
E
F
G
H

***Figura 38** – Execução do algoritmo mean-shift no trabalho de Sun, para o rastreamento de um brinquedo em um vídeo. A) Brinquedo adotado como objeto alvo. B) Indicação das caixas que identificam a posição no frame anterior e no frame corrente. C) Rastreamento no frame 52 do vídeo. D) Rastreamento no frame 70 do vídeo, com oclusão parcial do brinquedo. E) Rastreamento no frame 137 do vídeo. F) Rastreamento no frame 173 do vídeo. G) Rastreamento no frame 227 do vídeo. H) Rastreamento no frame 254 do vídeo. Fonte: Baseado nos resultados obtidos por Sun (2012).*

5.5.2. APRENDIZADO DE MÁQUINA E APRENDIZADO PROFUNDO

A evolução tecnológica atual, bem como a disponibilidade e facilidade de acesso a recursos computacionais avançados tornou rotineira a utilização de ferramentas que possuem inteligência artificial embutida. O termo inteligência artificial se refere a algoritmos que realizam qualquer tipo de tarefa que necessita de certo nível de raciocínio, sendo esta capacidade associada a inteligência humana. Dentre as diversas ramificações da inteligência artificial temos o aprendizado de máquina, em inglês *machine learning*, que representa o conjunto de algoritmos que buscam aprender e reconhecer padrões automaticamente a partir de dados previamente apresentados. Os conceitos referentes a classificadores lineares e algoritmos de *tracking,* exibidos e aplicados anteriormente neste projeto, são exemplos de algoritmos de *machine learning*. Outra classe de algoritmos pertencente ao aprendizado de máquina são as redes neurais artificiais que possuem sua estrutura inspirada no cérebro humano, possuindo diversas interconexões entre os neurônios (SZE et al, 2017). A Figura 39 ilustra a relação existente entre a inteligência artificial, o aprendizado de máquina e as redes neurais artificiais.

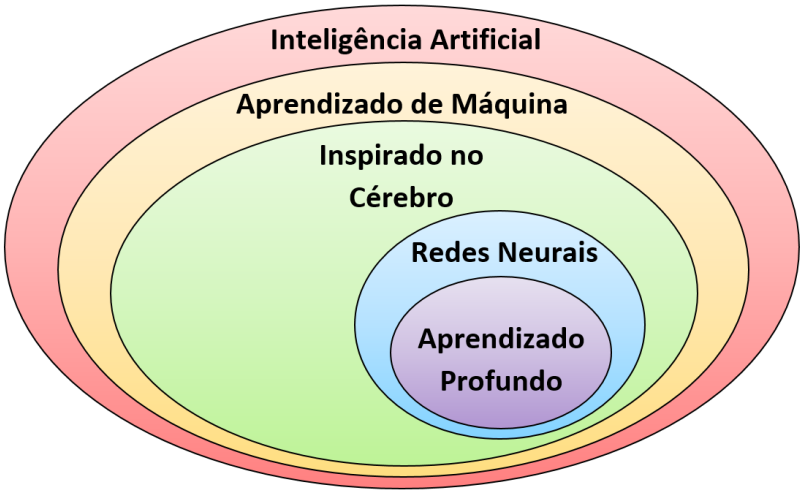


***Figura 39** – A inteligência artificial possui o aprendizado de máquina como um de seus subgrupos, enquanto este por sua vez engloba os algoritmos de redes neurais artificiais que podem ser agrupados no que é conhecido como aprendizado profundo, ou deep learning. Fonte: SZE (2017).*

As redes neurais artificiais tomam como base que os neurônios avaliam todos os estímulos de entrada, realizando uma soma e ponderação de seus pesos, para definir qual será a informação propagada adiante (SZE et al, 2017). Desta forma, um neurônio artificial recebe uma ou mais

entradas, cada uma com um peso específico, que são somadas e submetidas a uma função de ativação que define se o sinal é transmitido adiante ou cessado. Podemos associar diversos neurônios para receber a mesma informação de entrada, formando o que é conhecido como camada de neurônios, e a associação de três ou mais camadas dá origem uma rede neural de aprendizado profundo. A técnica de aprendizado profundo, ou *deep learning*, engloba as redes neurais convolucionais (*convolutional neural network* – CNN) denominadas dessa forma pois possuem camadas específicas que realizam a operação de convolução entre o dado de entrada e a matriz de pesos onde reside de fato o aprendizado da rede. A passagem de uma imagem através de uma rede CNN faz com que as dimensões originais sejam reduzidas enquanto a quantidade de neurônios, ou canais de recursos, por camada aumenta para que assim os atributos discriminatórios entre as classes sejam obtidos. A camada precedente da camada de saída, onde são informadas as probabilidades para cada uma das classes, é denominada como camada *fully connected* devido a forma de conexão entre os neurônios e corresponde a etapa da rede, onde de fato ocorre a classificação com base nos recursos extraídos previamente. A arquitetura genérica de uma CNN é mostrada através da Figura 40.

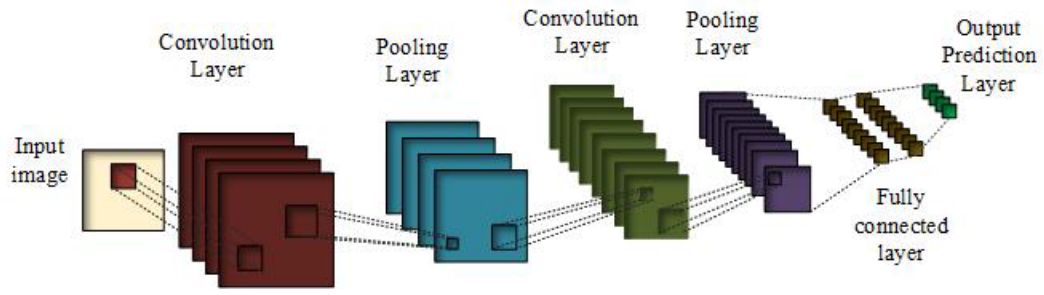


***Figura 40** – Arquitetura genérica de uma rede neura convolucional contendo a camada de entrada (input image), camadas convolucionais (convolution layer) seguidas por camadas de amostragem (pooling layer) que reduzem a dimensão dos mapas de atributos e aumentam a quantidade de neurônios. A camada fully connected é responsável pela classificação de fato, gerando probabilidades que são apresentadas na camada de saída (output prediction layer). Fonte: MUNIR (2019).*

5.5.3. U-NET: REDES COMPLETAMENTE CONVOLUCIONAIS

As arquiteturas de rede denominadas como redes completamente convolucionais (*fully connected network* – FCN) são encontradas na literatura atual como o estado-da-arte para tarefas de segmentação. A principal particularidade deste tipo de arquitetura, se comparada à CNN, reside no fato de que a camada final de classificação é substituída por camadas convolucionais (LONG et al, 2017). Dentre as diferentes configurações existentes de FCN, a estrutura U-Net proposta por Ronneberger e colaboradores em 2015, evidenciada pela Figura 41, se mostra como um algoritmo promissor para segmentações rápidas e precisas de imagens. A nomenclatura U-Net se deve ao fato de que essa arquitetura apresenta duas etapas principais de processamento, identificadas como caminho de contração e caminho de expansão, que apresentam simetria entre si e por isso geram um formato em U (RONNEBERGER et al, 2015).

O caminho de contração consiste na estrutura clássica de uma CNN, possuindo as operações de convolução seguida por normalização e obtenção dos recursos através de uma função de ativação, sendo esta normalmente a unidade de retificação linear (ReLU). A operação subsequente a ser realizada é a de amostragem através de maxpooling, onde ocorre a redução das dimensões da imagem. Desta forma, no decorrer do caminho de contração o mapa de recursos apresenta aumento na quantidade de canais enquanto as dimensões da imagem são reduzidas (ALOM et al, 2019). Em contrapartida, o caminho de expansão busca reconstruir as dimensões originais da imagem de entrada enquanto produz mapas de segmentação com base no mapa de recursos extraídos no caminho de contração (NGUYEN et al, 2017). O aumento das dimensões da imagem é realizado através da operação de convolução de transposição, onde o

mapa de recursos corrente é concatenado com o mapa de recursos correspondente no caminho de contração, sendo então interpolados (RONNEBERGER et al, 2015; NGUYEN et al, 2017). O resultado da interpolação equivale a um novo mapa de recursos, que deve ainda ser submetido a uma normalização e uma função de ativação de forma que o mapa se segmentação seja obtido.

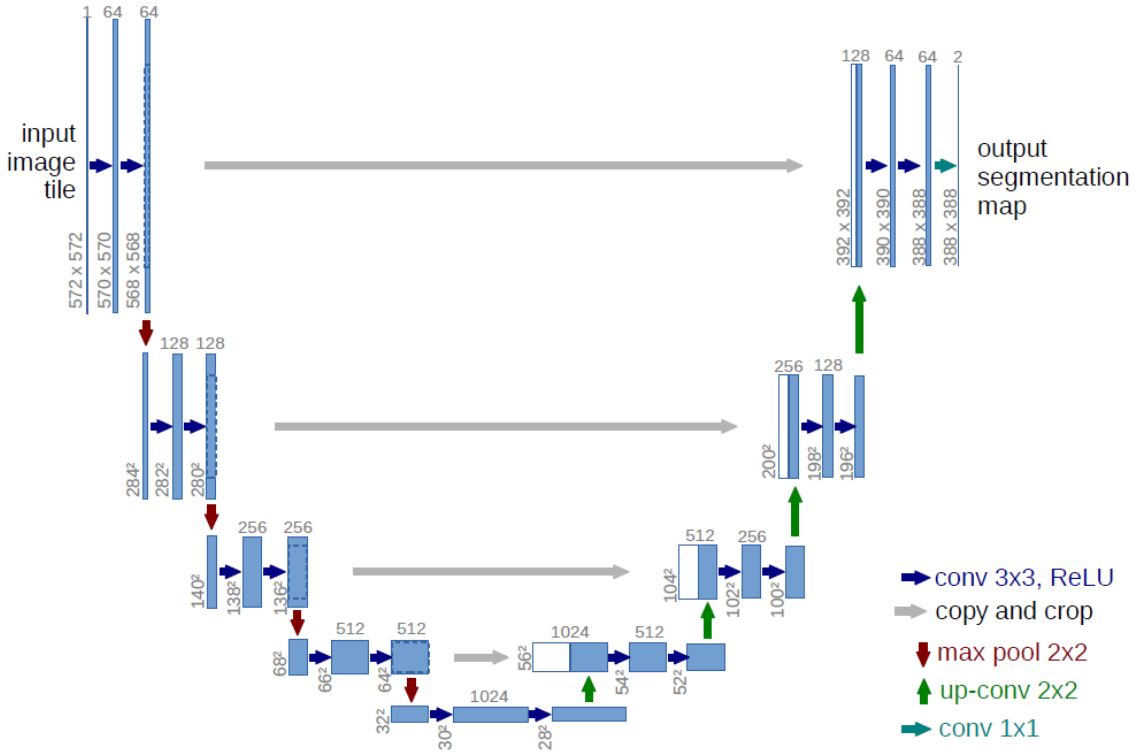


Figura 41 *– Arquitetura de rede U-Net original apresentada por Ronneberger et al. As caixas azuis indicam os mapas de recursos multicanais, sendo o número em sua parte superior o número de canais (neurônios). As caixas brancas representam os mapas de recursos copiados do caminho de contração e sua concatenação no caminho de expansão é dada pela seta cinza. As setas azuis correspondem as operações de convolução, seguido por normalização e função de ativação ReLU. A operação de maxpooling, responsável por reduzir as dimensões da imagem, é representada pela seta vermelha. A seta verde corresponde a operação de convolução de transposição (up-convolution), que realiza o aumento das dimensões da imagem. Fonte: RONNEBERGER (2015).*

A arquitetura **U-Net** permite uma varredura das imagens com kernels de tamanhos pré-estabelecidos, considerando ainda que o número de camadas existentes na rede influencia diretamente na dimensão mínima que as imagens precisam apresentar. Devido a forma de processamento das imagens completas, a **U-Net** permite uma segmentação com a localização global dos objetos ainda que preserve o contexto das imagens de entrada. Destaca-se que a U-Net é potencialmente eficiente mesmo para *datasets* que apresentam pequenas quantidades de dados (ALOM et al, 2019) como é o caso do trabalho realizado por Tseng & Lin em 2020, onde a métrica mIoU (*mean intersection over union*) atingiu nível superior a 76% mesmo utilizando um conjunto de somente 400 imagens.

Foram percebidas limitações no processo de ***Machine Learning***, através do treinamento por **SVM**, principalmente quanto ao custo computacional, denotando gargalos no processamento paralelo de dois sonares em tempo real na solução proposta neste projeto. A falta de melhora da acurácia da solução, mesmo com o aumento do número de treinamento, principalmente quanto a detecção de peixes menores, apresenta-se como uma restrição para o uso da **SVM** nesta aplicação. Optou-se por tanto, pela alteração da arquitetura do modelo de inteligência artificial, passando de ***Machine Learning*** para ***Deep Learning*****,** realizando novos treinamentos com **CNNs** como a **U-NET** para a identificação e segmentação dos peixes. O uso desta arquitetura supera os obstáculos enfrentados pela **SVM,** uma vez que possui custo de processamento inferior, permitindo o processamento idealizado em tempo real, e pelos resultados de acurácia apresentados, superiores mesmo para grupos pequenos de imagens de treinamento. Esta

mudança afetará apenas no algoritmo de detecção dos peixes, mantendo-se todas as demais funcionalidades da aplicação e tracking para a contagem, buscando justamente a otimização da solução e molhara de performance, e precisão.

5.5.4. ARQUITETURA YOLO

Como apontado na entrega anterior, arquiteturas de redes neurais denominadas como redes completamente convolucionais (*fully connected network* – FCN) são atualmente encontradas na literatura como o estado-da-arte para tarefas de segmentação e identificação de objetos. A principal particularidade destas arquiteturas, se comparada à CNN, reside no fato de que a camada final de classificação é substituída por camadas convolucionais (LONG et al, 2017). Dentre as diferentes configurações, havíamos trabalhado a princípio com a estrutura U-Net proposta por Ronneberger e colaboradores em 2015, evidenciada na Figura 41, que se mostrou como um algoritmo promissor para segmentações precisas das imagens, e obteve resultados animadores para o aumento da acurácia na identificação dos peixes em nossa aplicação. Contudo, uma vez que a U-Net permite uma varredura das imagens com kernels com tamanhos pré-estabelecidos, a fim de identificar e segmentar os objetos de interesse, ainda possui um custo computacional levemente elevado, inferior as metodologias de Machine Leanirg, mas ainda pesado para o processamento em tempo real. Assim, uma alternativa nesta linha de desenvolvimento é a arquitetura YOLO: You Only Look Once.

Proposta em 2015 por Joseph Redmon e Ali Farhadi, o YOLO foi logo reconhecido como uma técnica inovadora pois através de uma abordagem totalmente nova foi capaz de obter uma precisão igual ou superior ao dos outros métodos de detecção de objetos da época, porém com uma velocidade de detecção muito superior. Enquanto as técnicas mais precisas da época levavam aproximadamente 0.5 segundos (ou mais) para processar uma imagem, a YOLO conseguia detectar com o mesmo nível de precisão em menos de 0.05 segundos, o que permitiu ser utilizado em aplicações de tempo real, sendo capaz de rodar até em uma taxa de 30 frames por segundo. Outro motivo do sucesso do YOLO é o fato de ser totalmente código aberto e livre de licenças de uso. Ou seja, desde o código fonte, até a arquitetura da rede neural e os pesos pré-treinados, tudo isso pode ser usado por qualquer um e de qualquer forma. Atualmente, essa técnica é considerada o estado da arte em detecção de objetos em tempo real. Em abril de 2020, quase 5 anos após o lançamento de sua primeira versão, o YOLO tem a sua quarta versão oficialmente publicada. Até o momento da data de sua publicação, o YOLOv4 é o detector de objetos com maior acurácia que permite ser executado em tempo real para a detecção d e inúmeros objetos, similares ou distintos, como pode ser visto na Figura 42.

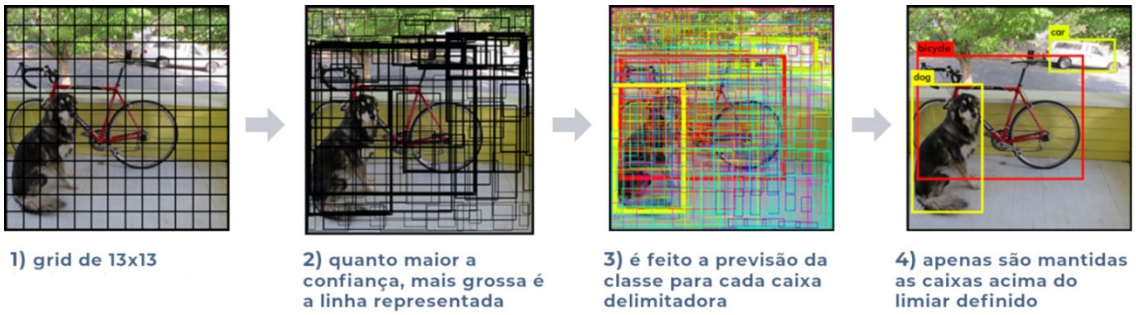


Figura 42 *– Detecção de objetos a partir da arquitetura de rede U-Net Yolo. 1. Primeiro, o algoritmo divide a imagem em um grid de S x S células. 2. Cada uma dessa células é responsável por fazer a predição de 5 caixas delimitadoras (B), para caso haja mais de um objeto naquela célula. Também é retornado à pontuação de confiança que nos diz o quanto de certeza ele tem que aquela caixa delimitadora contenha um objeto. 3. Para cada caixa, a célula também faz a previsão de uma classe. Isso funciona como se fosse um classificador: é fornecido um valor de probabilidade para cada uma das classes possíveis. O valor de confiança para a caixa delimitadora e a predição da classe são combinados em uma pontuação final, que vai nos dizer a probabilidade dessa caixa conter um objeto específico. 4. Acontece que a maioria dessas caixas terá um valor de confiança extremamente baixo, então por isso geralmente se considera apenas as caixas cuja pontuação final seja 30% ou mais. Esse valor de 30% é o limiar, chamado de threshold, e ele pode ser alterado dependendo do quão preciso você quer que o detector seja.*

Diferente da classificação, que busca apenas prever a classe presente na imagem, a detecção de objetos além de prever qual é a classe precisa identificar também a localização do objeto nessa imagem. Por exemplo, quando construímos um classificador de cachorro e gatos, o nosso classificador vai retornar como resultado da predição a classe (cachorro ou gato) e a confiança (grau de certeza sobre aquela predição). Mas e se a imagem tiver tanto o gato quanto o cachorro? Poderíamos talvez treinar classificadores multi-classe que vão ser capazes de identificar as duas classes (tanto cachorro como gato). No entanto, nós ainda não temos como saber a localização do cachorro ou gato na imagem. Esse problema de identificar a localização de um objeto (dada a classe) em uma imagem é chamado de *localization* (localização), que é justamente o retângulo que contém a localização de cada objeto identificado (*bounding box*) conforme pode ser visto na Figura 42. Ser capaz de predizer a localização do objeto na imagem junto com a classe do objeto (para múltiplas classes) é o que chamamos de Detecção de Objetos. E é justamente através desta metodologia, que permite o processamento em tempo real com alta precisão, que aumentaremos a acurácia e performance da solução proposta neste projeto, visando a identificação e contagem dos peixes nas sub-regiões definidas.

A utilização de redes neurais de aprendizado profundo em tarefas de identificação e segmentação de objetos permite alcançar alta acurácia, porém é necessário considerar outros parâmetros envolvidos nesse tipo de processamento. O principal aspecto que deve ser analisado quando o reconhecimento dos padrões deve ser feito em tempo real é o tempo decorrido entre a apresentação de um frame e a classificação de saída na imagem. O processo de extração de características da imagem, através da aplicação de consecutivas convoluções, pode tornar custosa e lenta a utilização de CNNs como as arquiteturas AlexNet e VGG (HUANG et al, 2020). Sendo assim, a utilização de arquiteturas de rede que processam a imagem por completo de uma única vez mostram-se como uma alternativa viável no contexto de velocidade de classificação. Dentre os diferentes tipos de arquitetura que apresentam essas características pode-se citar a YOLO (REDMON et al, 2016), YOLOv2 (REDMON et al, 2017) e YOLOv3 (REDMON et al, 2018).

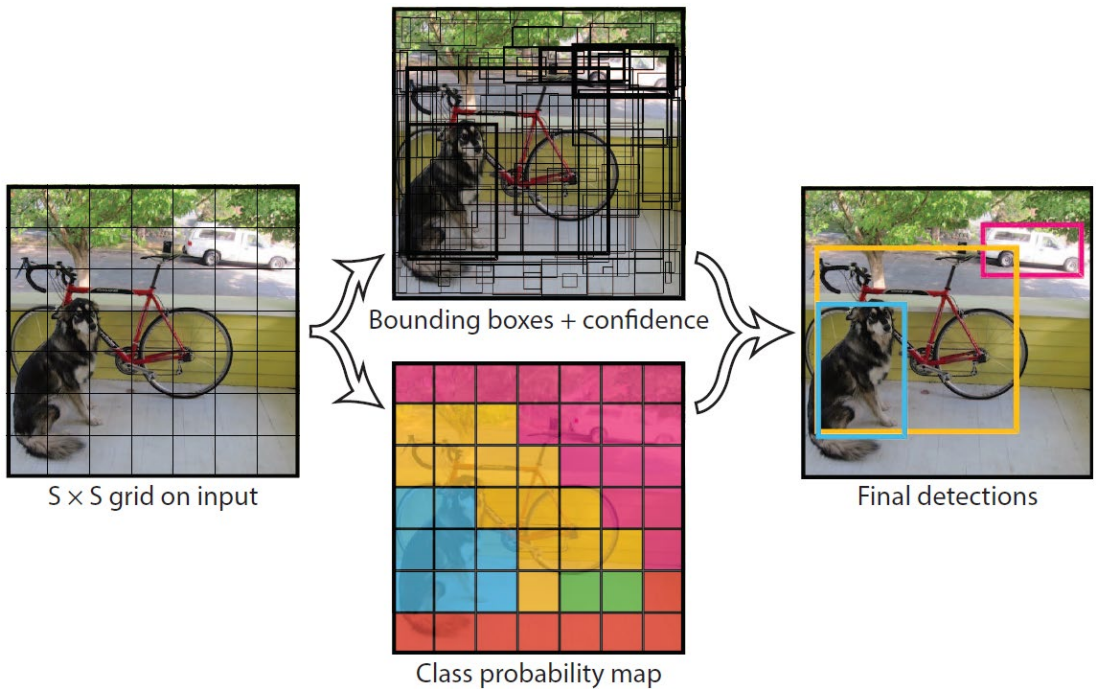


Figura 43 *– Diagrama esquemático da forma de processamento da imagem em uma arquitetura de rede YOLO. De maneira genérica, a imagem completa é subdividida em um conjunto de células que são analisadas individualmente com relação a todas as classes de acordo com o nível de confiança e as probabilidades calculadas. Fonte: REDMON (2016).*

A arquitetura de rede YOLO é denominada dessa forma devido o conceito base onde a imagem é olhada uma única vez (em inglês: *you only look once* – YOLO), sendo capaz de identificar os objetos presentes na imagem e suas posições (REDMON et al, 2016). Essa configuração de rede faz parte de um projeto denominado como DarkNet, que reúne códigos livres desenvolvidos em C voltados para redes neurais (DarkNet, 2021). Uma imagem de entrada aplicada na rede YOLO é subdividida em uma grade de células com dimensão SxS, como mostra a Figura 42, sendo cada uma dessas células analisadas individualmente. Cada uma das células é responsável por determinar B caixas de predição, que indicam a presença de determinada classe naquela região, junto a uma pontuação de confiança para a presença daquela caixa e uma estimativa de probabilidade. É relevante destacar que cada uma das células permite o retorno de uma única probabilidade, independentemente do número de caixas de predição identificadas. Após o processamento de cada célula, uma combinação entre as caixas de predição e as probabilidades é realizada de forma a obter estatisticamente a caixa de predição que mais representa a região.

5.5.5. AJUSTES E RECORTES

Uma segunda alteração consiste na redefinição dos cortes da tela do software do sonar para a extração das imagens para o processamento e identificação dos peixes. A primeira fase consistiu na identificação dos 2 sonares, e o recorte uniforme a partir do centro do software do sonar, como pode ser visto na Figura 43, onde em (a) possuímos a tela inteira do software e em (b) a repartição recortada quanto a aquisição das imagens dos sonares. Em seguida foram recortados os retângulos referentes as margens dos canais em ambos os sonares definindo as paredes laterais do canal, e delimitando as sub-regiões de avaliação, conforme a Figura 44. Nesta figura, já é também possível avaliar a detecção de objetos que serão repassados para o modelo da Yolo a partir do ajuste do ruído nos canais azul e verde da imagem, para o isolamento dos peixes em relação ao ruído, através das operações morfológicas tratadas no fluxograma da Figura 45.



Figura 44 *– (a) Tela do Software do sonar. (b) Extração da tela dos dois sonares para o processamento e aquisição das imagens.*

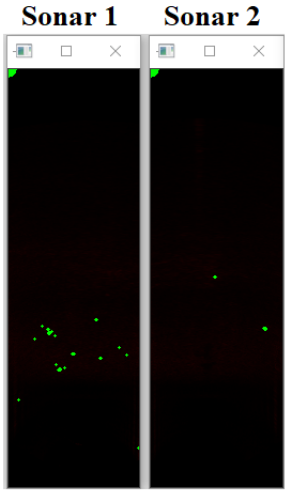


Figura 45 *– Recortes das paredes dos 2 sonares referentes a delimitação da análise das imagens. Imagens já com o pré-processamento para eliminação do ruído pelos ajustes nos canais azul e verde e operações morfológicas.*

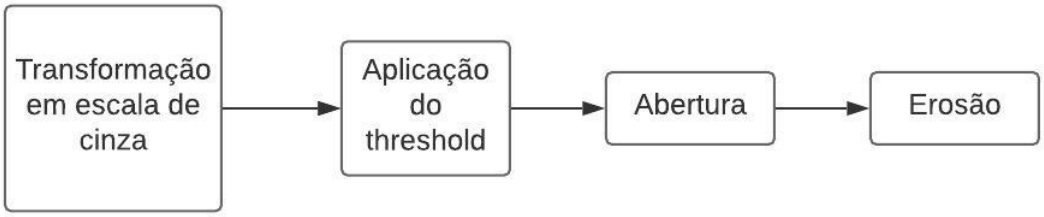


Figura 46 *– Sequência de operações morfológicas após o ajuste do ruído nos canais azul e verde das imagens. Transformação das imagens em escala de cinza. Aplicação do threshold Otsu. Operações de abertura e fechamento com elementos binários.*

5.5.6. MODIFICAÇÕES DA ETAPA DE PRÉ-PROCESSAMENTO

O registro da atividade da ictiofauna no interior do tubo de sucção através do sistema SONAR, como mostrado anteriormente nos capítulos 5.4.3 e 5.4.4, possui dependência direta no comportamento das ondas de ultrassom emitidas pelo equipamento. As imagens acústicas caracteristicamente apresentam um ruído conhecido como *Speckle,* responsável por dar um aspecto granular à imagem e dificultar a separação dos objetos presentes. A secção 5.3.6 deste projeto abordou superficialmente a interferência desse tipo de ruído (Figura 25) e um novo exemplo é apresentado pela Figura 43, que mostra um dos frames registrados pelo SONAR, caracterizando e tornando evidente o ruído *Speckle*.

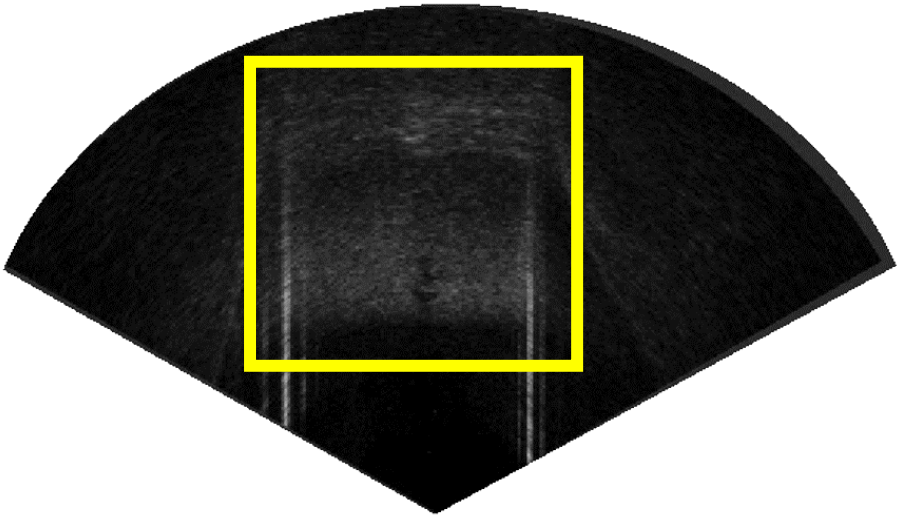


***Figura 47** – Registro acústico realizado pelo sistema SONAR com destaque em amarelo para a região com maior ocorrência de ruído Speckle, sendo essa interna a área de movimentação dos peixes. Fonte: Elaborado pelos autores.*

A literatura apresenta diversas alternativas para a remoção ou atenuação desse tipo de ruído, sendo tipicamente utilizados filtros específicos que possuem essa finalidade. Entretanto, uma análise minuciosa dos canais de cores que representam cada imagem mostrou maior simplicidade na resolução deste problema. Os registros acústicos do SONAR são compostos por imagens constituídas por três canais de cores, conhecidos como canais RGB (*red, green, blue),* que carregam em si informações referentes a construção da imagem. O conteúdo dos canais de cores na imagem acústica não possui grande relevância, considerando que cada frame em sua versão em tons de cinza permite a observação das informações sem perdas. Analisando separadamente cada um dos canais foi observado que os canais G e B, que representam respectivamente as componentes verdes e azuis da imagem, carregam consigo grande parte do ruído *Speckle.* Desta forma, considerando que a eliminação desse ruído permite a identificação dos peixes mais facilmente, ambos os canais foram eliminados e somente o canal R foi mantido para os processamentos de imagem subsequentes.

5.5.7. RESULTADOS

5.5.7.1. METODOLOGIA FINAL

Durante o desenvolvimento do presente projeto diversas melhorias e alterações foram realizadas na metodologia de identificação e rastreamento dos peixes, visando a obtenção de melhores resultados. Os registros acústicos obtidos através do sistema SONAR devem ser processados *frame* a *frame*, uma vez que em sua aplicação final o software utilizará as imagens obtidas em tempo real. Sendo assim, a primeira etapa de desenvolvimento consiste no isolamento de cada *frame* bem como a remoção da região de interesse de análise, que corresponde à área que pertence ao interior do tubo de sucção e onde a movimentação de peixes ocorre. O resultado do recorte do frame possui formato de canais de cores RGB, forma como os registros são realizados, e devido as análises prévias realizadas, somente o canal vermelho (R) deve ser mantido e os demais devem ser removidos a fim de minimizar os efeitos do ruído *Speckle* na correta identificação dos peixes. A imagem resultante irá conter *pixels* com intensidades em escala de cinza, estando então preparada para a aplicação do filtro bilateral, da transformação de intensidade e binarização como na metodologia apresentada pela secção 5.3.5. A etapa final da fase de pré-processamento é composta pela determinação dos contornos

que definem os objetos presentes na imagem seguida pela aplicação das operações morfológicas de abertura e erosão, com o intuito de separar contornos em contato direto e eliminar possíveis ruídos ainda existentes.

A execução das etapas de pré-processamento das imagens acústicas é essencial para a remoção de informações que são irrelevantes para a identificação automática dos peixes. Sendo assim, com a permanência somente desses contornos na imagem pode-se então realizar o treinamento e teste da arquitetura de rede YOLO. Considerando todos os registros acústicos disponíveis, foram extraídos um total de 2.000 (dois mil) frames cada um contendo dois focos do interior do tubo de sucção (Figura 44) resultando, portanto, em um *dataset* contendo 4.000 imagens com peixes anotados manualmente. O conjunto de imagens disponíveis foi então dividido entre conjunto de treinamento e conjunto de teste, onde as imagens do primeiro conjunto são utilizadas para o treinamento da rede enquanto o segundo é utilizado somente para a avaliação do desempenho da arquitetura construída. Neste projeto foram destinados 80% do *dataset* para o conjunto de treinamento e 20% para o conjunto de teste.

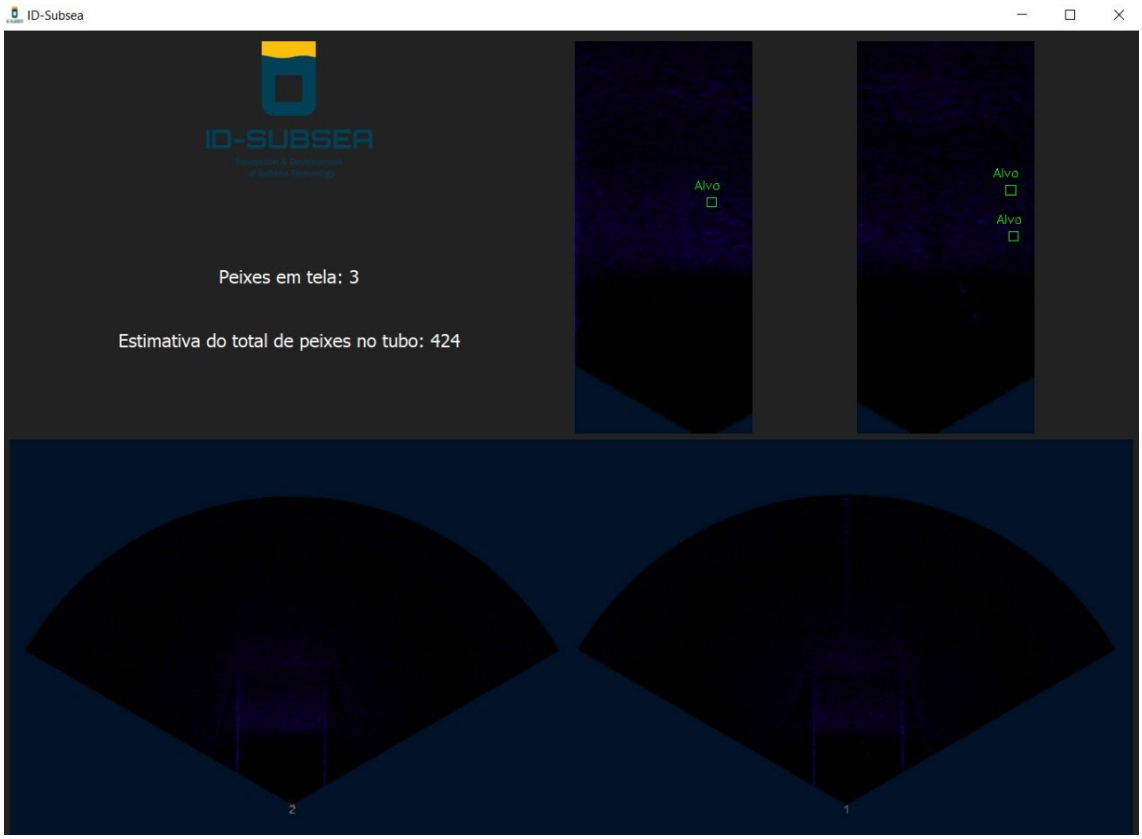


Figura 48 *– Tela da solução elaborada com a realização dos recortes das seções em ambos os tubos com a contagem de peixes que surgem na tela e a estimativa do total que permanece dentro dos tubos.*

A identificação dos objetos que representam os peixes nas imagens, através da YOLO, permite então que o algoritmo de rastreamento seja aplicado, nesse caso o algoritmo de Camshift já apresentado foi o utilizado na aplicação final. O rastreamento dos peixes é essencial para acompanhar a movimentação dos animais no interior do tubo de sucção, permitindo observar se a trajetória tomada tende em direção às turbinas ou rumo à saída do canal. Na Figura 44 é possível avaliar tanto a identificação dos peixes pela rede neural, que estão demonstrados pelos boxes verdes, e a contagem de peixes dentro do tubo pela avaliação de deslocamento vetorial dos peixes que são identificados na tela, armazenando o retirando da contagem total dependendo se há movimentação para dentro ou para fora do tubo.

5.5.7.2. RESULTADOS DO TREINAMENTO

Como apontado no item anterior, para o treinamento e validação da rede neural com arquitetura YOLO proposta para este projeto, o dataset de 4.000 imagens de frames dos sonares com peixes marcados manualmente foi dividido em 80% para treinamento (3.200 imagens) e 20% para teste (800 imagens). O número de camadas de treinamento, filtros, e os otimizadores lineares utilizados na versão final foram obtidos pela avaliação exaustiva do treinamento da rede neural, escolhendo o modelo que apresentou maior valor de acurácia. E essa métrica foi obtida pela avaliação do número de peixes identificados justamente nas 800 imagens de teste, que correspondiam a um total de 1.920 peixes marcados manualmente (*Gold Standard*). Após inúmeros treinamentos e testes realizados, a rede neural que obteve maior assertividade alcançou cerca de 90% de acurácia. A Tabela 1 abaixo apresenta o número de Verdadeiros Positivos (VP), ou seja, casos que a rede identificou e realmente era peixes marcados; Falsos Positivos (FP), casos em que a rede identificou, mas não eram peixes; Falsos Negativos (FP), casos em que a rede deixou de identificar peixes marcados; e os valores de Acurácia e Sensibilidade calculados para esse modelo neural. E na Figura 49 é possível avaliar na tela do software desenvolvido a identificação correta de dois peixes no frame demonstrado (boxes em verde). Já na Figura 45 b) é possível avaliar que o modelo neural identificou um peixe (box verde) e deixou de identificar um segundo presente na imagem (círculo vermelho).

	VP	FN	FP	Acurácia	Sensibilidade
Gold Standard	1920	0	0	100%	100%
Modelo Neural	1785	76	65	89,9%	93,4%

Tabela 1: *Avaliação do modelo neural YOLO para identificação dos peixes nas imagens dos frames do sonar. Total de 800 imagens no conjunto teste com 1920 peixes marcados manualmente.*

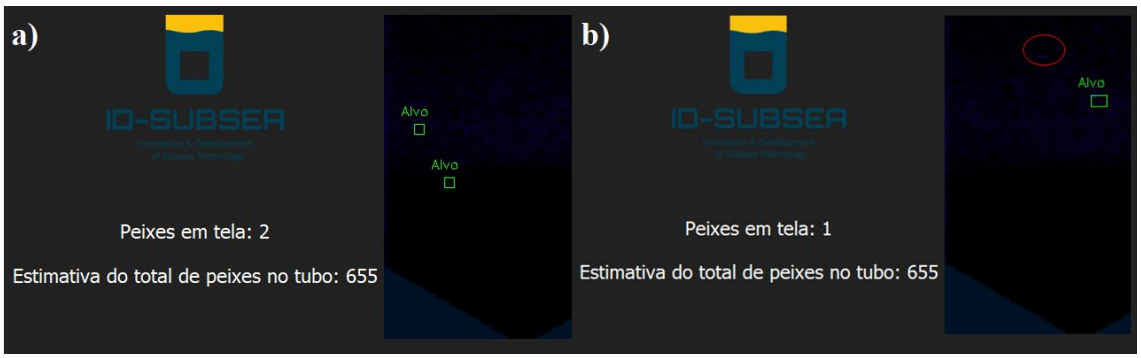


Figura 49 *– Avaliação do resultado de teste do modelo neural de identificação dos peixes. a) Frame onde foi detectado corretamente dois peixes anotados manualmente (boxes em verde). b) Frame onde é detectado corretamente um peixe (box verde) e não é identificado um segundo peixe na tela (círculo vermelho).*

5.5.7.3. CONCLUSÕES TREINAMENTO

Os valores apresentados na Tabela 1 demonstram a robustez e eficiência do modelo neural de identificação dos peixes. A taxa de falso negativo é de cerca de 8% (65 em 800 imagens) demonstrando que a solução não pega muitos ruídos ou artefatos como se fossem peixes. Já em relação a sensibilidade, foco principal do projeto, a solução alcançou 93,4%, sendo

superior a muitos resultados encontrados na literatura. Estes valores demonstram o quanto a solução chegou ao potencial de identificar os peixes conforme as anotações de treinamento, e que o modelo deverá ser aplicado em casos reais para avaliação do comportamento d e identificação e contagem. Inclusive, caso a taxa de não identificação dos peixes permanecer similar ao do treinamento e validação, cerca de 6,6%, é possível cria rum mecanismo de aproximação em relação ao valor de peixes identificados para estimar o valor mais próximo ao real de peixes no tubo.

5.5.8. MEDIÇÕES

5.5.8.1. AFERIÇÕES NA USINA

Entre os dias 16/06/2021 e 21/06/2021, foram realizados testes na usina para avaliar o desempenho do algoritmo de identificação e contagem de peixes utilizando a solução em campo, justamente aferindo as imagens dos sonares nos dois tubos onde estão instalados a fim de monitorar e calcular o número de peixes que estariam no tubo durante a realização dos testes. Neste período, foram realizados dois testes: I) Sem o uso da barreira de contenção e II) Com o uso da barreira de contenção para avaliar também a eficácia da mesma. Para ambos os testes, houve a parada das máquinas 24 horas, realizando-se após esse período a secagem e contagem manual do número de peixes que ficaram presos dentro do tubo. Os resultados da contagem automática pela solução e algoritmo de inteligência e a contagem manual estão presentes na Tabela 2 a seguir, assim como a sensibilidade da solução.

	Contagem Modelo Neural	Contagem Manual	Sensibilidade
Teste I: Sem barreira de contenção	2253	8208	27,4%
Teste II: Com barreira de contenção	424	694	61,0%

Tabela 2: *Resultados da aferição da contagem de peixes pela solução e de forma manual nos testes realizados na usina com e sem o uso da barreira de contenção.*

5.5.8.2. ANÁLISES APÓS AFERIÇÕES NA USINA

Após a realização dos testes na usina, foi realizada avaliação e análise dos resultados a fim de identificar os problemas do estudo de caso em campo, que acabou apresentando valores de sensibilidade discrepantes aos obtidos na validação do treinamento da solução. Com o uso das filmagens salvas do sonar que foram gravadas nos testes, o primeiro ponto levantado foi o dimensionamento de algumas marcações que o modelo neural acabou identificando como um único peixe. Na Figura 46 (a) é possível avaliar a presença de uma mancha de movimentação relativamente grande no sonar, e realizando a ampliação e mensuração desse achado (Figura 46 (b)) é possível analisar que o achado tem cerca de 1,8 metros em sua maior extensão. Com a suspeita de não haver peixes com tal comprimento adentrando a sucção , e pelo comportamento de movimentação desse e outros achados similares, no 5º ensaio de efetividade, e pela referência de tamanho médio dos peixes estão presentes nas tabelas 1 e 2 da Figura 47. Através da análise desses valores, observamos que as médias de tamanho das espécies com maior ocorrência na contagem dos testes são de 18 e 29 cm, valores muito abaixo das manchas

de quase 2m registradas. Concluindo assim a possibilidade de serem cardumes de peixes, que estão sendo mensurados uma única vez pelo algoritmo.

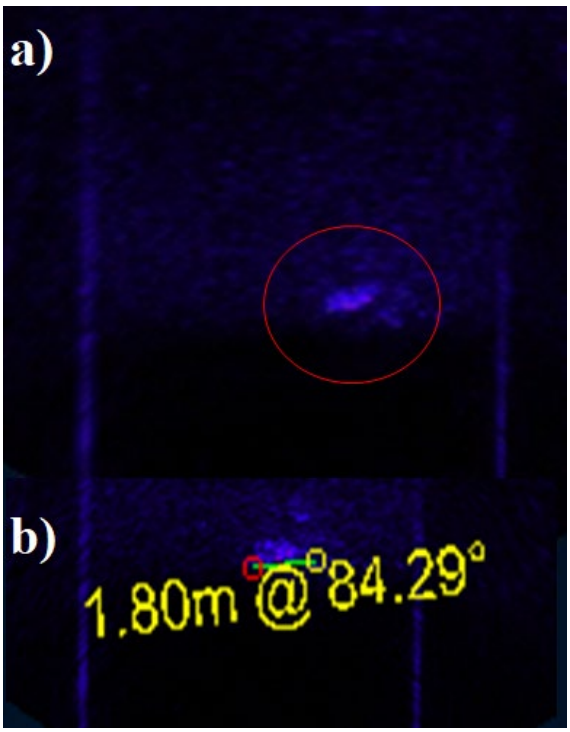


Figura 50 *– Identificação de achados no sonar de maior volume que podem representar cardumes ao invés de peixes individuais. a) Frame original do sonar com a marcação do achado. b) Achado ampliado e mensurado em 18 metros de comprimento.*

Tabela 1. Número de peixes resgatados no tubo de sucção (TS) e no poço de esgotamento (PE) da UG3 durante o 5º ensaio de eficiência da barreira de repulsão de peixes, realizado entre os dias 16 e 21 de junho de 2021 na UHE Machadinho.

Nome científico	Nome popular	Barreira desligada			Barreira ligada		
		TS	PE	Total	TS	PE	Total
Apareiodon affinis	canivette	-	2	2	-	-	0
Bryconamericus stramineus	lambari	3	6	9	-	-	0
Crenicichla missioneira	Joana	-	1	1	-	-	0
Eigenmannia virescens	Tuvira	-	5	5	-	-	0
Hydrolycus scomberoides	peixe-cachorro	1	-	1	-	-	0
Hypostomus luteus	cascudo	12	3	15	-	-	0
Iheringichthys labrosus	mandi-beiçudo	252	7498	7750	-	679	679
Oligosarcus brevioris	peixe-cachorro	1	-	1	-	-	0
Parapimelodus valenciennis	mandi	-	3	3	-	-	0
Pimelodus maculatus	pintado-amarelo	281	139	420	4	11	15
Serrasalmus maculatus	piranha	-	1	1	-	-	0
Total de peixes (*n*)				8208			694
Eficiência (%)				91,5			
Biomassa (kg)		100,0	253,1	353,1	1,2	20,8	22,0
Eficiência (%)				93,8			

Tabela 2. Comprimento total médio (± desvio padrão) do mandi-beiçudo e do pintado-amarelo resgatados no 5º ensaio de eficiência da barreira de repulsão de peixes.

Espécie	Comprimento total (mm)	Tamanho da amostra (*n*)
Iheringichthys labrosus	180,5 ± 25,2	204
Pimelodus maculatus	289,3 ± 58,1	143

Figura 51 *– Tabelas do relatório da XXX a respeito da avaliação do número de peixes resgatados nos testes com e sem o uso da barreira apresentado as espécies com maior ocorrência e os respectivos tamanhos médios.*

5.5.8.2. AJUSTES APÓS A ANÁLISE

Com a identificação da questão dos cardumes apresentada no ponto anterior, foram realizados ajustes para o aumento da sensibilidade do algoritmo de identificação e contagem dos peixes. Durante a demarcação manual dos peixes para o treinamento, foram feitas marcações individuais padrões por quadrados de 20 pixels de lado, que podem ter interferido a identificação de peixes menores, ou ainda na interpretação destes mesmos cardumes. E ainda, uma vez que estas marcações foram executadas de forma manual, podem ter sido marcados mais de um único peixe por fala de experiência dos responsáveis, diminuindo a sensibilidade de ajustes finos para a identificação e contagens de peixes muito próximos. Assim, foram realizados testes e alterações com os vídeos gravados nos testes de efetividade para o aumento da precisão da solução. Dentre os ajustes realizados foram a avaliação dos tamanhos dos achados, realizando a contagem por atividade e aproximação da contagem dos cardumes pelo tamanho médio dos peixes identificados. Além disso, foram realizados ajustes no ganho do sonar, variando a taxa de aquisição dos frames para o uso da solução, e o uso de novas marcações de peixes pequenos para aprimoramento da rede neural. Na Tabela 3, estão presentes a evolução dos resultados com a realização desses ajustes com ou uso da barreira.

Tabela 3: *Resultados após os ajustes na solução da sensibilidade na contagem do número de peixes nos testes realizados na usina com e sem o uso da barreira de contenção.*

	Contagem Manual	Contagem Modelo Neural Ajustado	Sensibilidade
Teste I: Sem barreira de contenção	8208	4650	56,6%
Teste II: Com barreira de contenção	694	558	80,0%

5.5.9. CONCLUSÕES

De acordo com a comparação dos resultados da tabela 2 e da tabela 3, houve significativa melhora na identificação e contagem dos peixes pela solução após o ajuste modelo e reconsideração e análise do tamanho dos achados. A sensibilidade do teste sem barreira passou de 27% para 56,6%, e do teste com barreira de contenção a sensibilidade melhorou de 61% para 80%. Esse avanço na sensibilidade da solução reflete o poderio de evolução que a solução possui, havendo melhoria apenas com os ajustes de avaliação do tamanho dos achados. Infelizmente, como não houve oportunidades ao longo do projeto em sua fase inicial de maiores testes e contagens de peixes para obtermos o padrão ouro da contagem real do número de peixes no canal (*Gold Standard*), sendo os *labels* marcados manualmente pela equipe de desenvolvimento sem experiência absoluta para compreender quando poderia haver mais de um único peixe em um agrupamento. De forma que a sensibilidade de treinamento foi muito mais alta, quase 93%, mas justamente considerando-se esse padrão de marcação. Além disso, o treinamento da rede foi baseado em identificação de peixes na imagem no sonar, e não houve avaliação anterior a respeito do número de peixes já no canal justamente por não haver tais dados para serem avaliados. Os resultados reais obtidos são inferiores aos obtidos durante a fase de P&D, e necessitam ainda de ajustes para poder disponibilizar a solução de forma operacional dentro do sistema da usina. Contudo, havendo apenas um teste de efetividade com a solução no local, e tendo os avanços apresentados com os ajustes parciais já realizados, é possível ainda realizar mais ajustes para alcançarmos os resultados pretendidos. As problemáticas enfrentadas não haviam sido previstas ao longo do projeto justamente pode essa falta de experiência e contato com a aplicação em campo, mas que agora após a realização da aferição, podem ser feitos os ajustes para tal objetivo.

5.5.9.1. PRÓXIMOS PASSOS

Como citado anteriormente, a melhora obtida nos resultados após os ajustes iniciais com os dados da aferição de teste mostra que é possível melhorar a solução desenvolvida, principalmente realizando o melhoramento da sub identificação de peixes agrupados e cardumes. Para isso faz se necessário a continuação do desenvolvimento e aperfeiçoamento do modelo neural e processamento das imagens, principalmente no rastreamento dos peixes, além da realização de mais testes de efetividade em campo para avaliar o uso da solução da rotina da usina. E uma alternativa para forçar ainda mais objetivo do projeto, é a avaliação de biomassa ao invés de apenas a quantidade de peixes, uma vez que pelo dimensionamento dos achados possa ser mais eficiente o controle de peso do fluxo de entrada e saída do canal. E por fim, poderemos adicionar ao projeto a parte de integração e dashboard de acompanhamento para disponibilizar a solução diretamente ao sistema da usina, e extração de estatísticas para o acompanhamento e monitoramento do fluxo de peixes no canal.

5.5.10. RECURSOS UTILIZADOS

Discorrer

5.5.11. MEDIÇÕES

Etapa	Atividade	Custo (R$)
10.1	Ap. Haar cascade	13.708,80
10.2	Criação 1º Script	19.347,12
10.3	Criação 2º Script	6.449,04
10.4	Criação 3º Script	10.748,40
10.5	Criação 4º Script	4.299,36
	TOTAL	**54.552,72**

Tabela 13 – Recursos Entrega 6

5.5.12. RECURSOS UTILIZADOS

Mês/Ano	Mat. De consumo	Mat. Perm e Equip.	Outros	Serv. 3os	Viagens e diárias	Total Mês
Set/20	0,00	0,00	0,00	0,00	0,00	0,00
Out/20	0,00	0,00	0,00	0,00	3.670,06	3.670,06
Nov/20	0,00	0,00	0,00	0,00	0,00	0,00
Dez/20	0,00	0,00	0,00	38.409,16	0,00	38.409,16
Jan/21	0,00	0,00	0,00	0,00	0,00	0,00
Fev/21	0,00	14.000,00	3.000,00	0,00	0,00	17.000,00

Tabela 14 – Reembolsos Entrega 6

6. BIBLIOGRAFIA

ALOM, Z.; YAKOPCIC, C.; HASAN, M. et al. Recurrent residual U-Net for Medical Image Segmentation. Journal of Medical Imaging, 6(1), 014006, 2019.

Amorim DD, Jussan C, Godinho AL, Loures RC. Avaliação de risco de morte de peixes em usinas hidrelétricas. Programa PeixeVIvo CEMIG. Belo Horizonte Cemig. 2016; Série Peixe Vivo, 5

Arvind CS, Prajwal R, Bhat PN, Sreedevi A, Prabhudeva KN. Fish Detection and Tracking in Pisciculture Environment using Deep Instance Segmentation. In TENCON 2019-2019 IEEE Region 10 Conference (TENCON). IEEE. 2019; pp. 778-783.

BRADSKI, G.R. Computer Vision Face Tracking for Use in a Perceptual User Interface. Intel Technol. J., 2(Q2), 1-15, 1998.

Clark D, Ruiz IT, Petillot Y, Bell J. Particle PHD filter multiple target tracking in sonar image. IEEE Transactions on Aerospace and Electronic Systems. 2007; 43(1), 409-416.

CHEN, W.; HU, X.; CHEN, W. et al. Airborne LiDAR Remote Sensing for Individual Tree Forest Inventory Using Trunk Detection-Aided Mean Shift Clustering Techniques. Remote Sensing, 10(7), 1-25, 2018.

DarkNet. Disponível em: https://pjreddie.com/darknet/. Acesso em: Dez/2020.

EMAMI, E. FATHY, M. Object Tracking Using Improved CAMShift Algorithm Combined with Motion Segmentation. Machine Vision and Image Processing, 7th, 1-4, 2011.

FEIJÓ, G.O. Um Algoritmo para o Rastreamento em Cardumes através da Análise de Imagens Digitais. Dissertação (mestrado) – Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre, 2016.

French G, Mackiewicz M, Fisher M, Challiss M, Knight P, Robinson B, Bloomfield A. JellyMonitor: automated detection of jellyfish in sonar images using neural networks. In 2018 14th IEEE International Conference on Signal Processing (ICSP). IEEE. 2018; (pp. 406-412).

FUKUNAGA, K.; HOSTETLER, L.D. The estimation of the gradient of a density function with applications in pattern recognition. IEEE Trans. Inf. Theory, 21(1), 32–40, 1975.

GODINHO, A.L.; LOURES, R.C. Avaliação de Risco de Morte em Peixes em Usinas Hidrelétricas. Capítulo 1: Risco de morte de peixes em usinas hidrelétricas, pág. 19-35. Companhia Energética de Minas Gerais, 2016.

GONZALES, R.C.; WOODS, R.E. Processamento Digital de Imagens. 3 Ed. Pearson, São Paulo, 2010.

HUANG, H.; SUN, D.; WANG, R.; ZHU, C.; LIU, B. Ship Target Detection Based on Improved YOLO Network. Mathematical Problems in Engineering, 1-10, 2020.

ImgLab. Disponível em: < https://imglab.in/>. Acesso em: 16 abr 2020.

KALMAN, R.E. A NewApproach to Linear Filtering and Prediction.Journal of Basic Engineering, 82, 35-45,1960.

LIU, J.; ZHONG, X. An object trackingmethod based on Mean Shift algorithm with HSV color space and texture features. Cluster Computing, 22(1), 6079-6090, 2018.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully Convolutional Networks for Semantic Segmentation. IEEE Trans Pattern Anal Mach Intell, 39(4), 640-651, 2017.

Lucas MC, e Baras E. Methods for studying spatial behaviour of freshwater fishes in the natural environment. Fish and Fisheries 1. 2000; 283-316.

MACHADO, A.F. Erosões e Dilatações Morfológicas Binárias Sequenciais Rápidas. Disponível em: <https://www.ime.usp.br/~dandy/mestrado.html>. Acesso em: 10 abr 2020.

MARTINS, L. Determinação do Fluxo Óptico em Dados Sísmicos. PUC-RIO. Disponível em: < http://webserver2.tecgraf.puc-rio.br/~mgattass/ra/trb08/LeonardoMartins/trab_final.html>. Acesso em: 27 jul 2020.

MUNIR, K. et al. Cancer Diagnosis Using Deep Learning: A Bibliographic Review. Cancers (Basel), 11(9), 1-36. 2019.

Nanonets. Introduction to Motion Estimation with Optical Flow. Disponível em:< https://nanonets.com/blog/optical-flow/>. Acesso em: 27 jul 2020.

NGUYEN, T. et al. Automatic phase aberration compensation for digital holographic microscopy based on deep learning background detection. Optics Express, 25(3), 15043-15057, 2017.

OPENCV. Cascade Classifier. Disponível em: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. Acesso em: 23 Jun 2020

Parker-Stetter SL, Rudstam LG, Sullivan PJ, Warner DM. Standard operating procedures for fisheries acoustic surveys in the Great Lakes. Great Lakes Fish. Comm, Spec. Pub. 2009; 180 pp.

REDMON, J.; DIVVALA, S.; GIRSHIK, R.; FARHADI, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 779-788, 2016.

REDMON, J.; FARHADI, A. “YOLO9000: better, faster, stronger” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6517–6525, IEEE, 2017.

REDMON, J.; FARHADI, A. “YOLOv3: an incremental improvement”. 2018. Disponível em: http://arxiv.org/abs/1804.02767.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention. Springer, Cham. 234-241, 2015.

Simmonds EJ, e Maclennan DN. Fisheries Acoustics: Theory and Practice, 2ed. Oxford: Blackwell. 2005; 413 pp.

SUN, J. A Fast Mean-Shift Algorithm-Based Target Tracking System. Sensors, 12(6), 8218-8235, 2012.

SZE, V.; CHEN, Y.H.; YANG, T.J. Efficient processing of deep neural networks: a tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329, 2017.

THEODORIDIS, S.; KOUTROMBAS, K. “Pattern Recognition”, 4ª Ed., Academic Press (Elsevier), 2009, Chapter 3 – Linear classifiers, 91 – 154.

TSENG, S.L.; LIN, H.Y. Fish Detection Using Convolutional Neural Network with Limited Training Data. Asian Conference on Pattern Recognition, 12046, 735-748, 2020.

Zhou C, Xu D, Chen L, Zhang S, Sun C, Yang X, Wang Y. Evaluation of fish feeding intensity in aquaculture using a convolutional neural network and machine vision. Aquaculture. 2019; 507, 457-465.

YAMASAKI, R.; TANAKA, T. Properties of Mean Shift. IEEE Transactions on Pattern Analysis and Machine Intelligence, 42(9), 2273-2286, 2019.

WEBB, A.R.; COPSEY, K.D. “Statistical Pattern Recognition”, 3ª Ed, John Wiley & Sons Ltd, 2011, Chapter 5 – Linear discriminant analysis, 221 – 274.