

INSTITUTO FEDERAL DE CIÊNCIA E TECNOLOGIA DE PERNAMBUCO
Campus Paulista
Curso Tecnólogo em Análise e Desenvolvimento de Sistemas

GABRIEL VAZ R. DOS SANTOS, ARTHUR GABRIEL PEDROSA DE L. CORREIA

**AVALIAÇÃO DE ALGORITMOS DE IDENTIFICAÇÃO DE IMAGENS PARA
MONITORAMENTO DE TUBARÕES(AIMT)**

Paulista
2025

GABRIEL VAZ R. DOS SANTOS, ARTHUR GABRIEL PEDROSA DE L. CORREIA

**AIMT - AVALIAÇÃO DE ALGORITMOS DE IDENTIFICAÇÃO DE IMAGENS PARA
MONITORAMENTO DE TUBARÕES**

Trabalho de conclusão de curso Monografia apresentada ao Departamento/ Programa/ Coordenação de Graduação em Análise e Desenvolvimento de Sistemas Instituto Federal de Ciência e Tecnologia de Pernambuco, como requisito para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.
Orientador: Prof. Dr. Anderson Apolinio Lira Queiroz

Paulista

2025

S237a Santos, Gabriel Vaz Rodrigues dos.

Avaliação de algoritmos de identificação de imagens para monitoramento de tubarões (AIMT) / Gabriel Vaz Rodrigues dos Santos; Arthur Gabriel Pedrosa de Lima Correia. – Paulista, PE: Os Autores, 2025.

26f. il. Color.

TCC (Curso Tecnólogo em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Pernambuco – Campus Paulista, 2025.

Inclui Referências

Orientador: Professor Dr. Anderson Apolinio Lira Queiroz

1. Monitoramento de tubarões. 2. Algoritmos de identificação de imagens. 3. Redes neurais convolucionais. 4. Aprendizagem de máquina. I. Título. II. Queiroz, Anderson Apolinio Lira (orientador). III. Instituto Federal de Pernambuco – Campus Paulista.

CDD 006.32

Catalogação na fonte: Bibliotecário Cristian do Nascimento Botelho CRB4/1866

RESUMO

Em meio aos ambientes naturais aquáticos, o convívio dos banhistas com os ataques de tubarões é bastante frequente. Assim, é notória a necessidade de um monitoramento mais efetivo, diante desses animais, a fim de garantir proteção e também promover mais segurança aos civis. Dessa forma, mediante a Aprendizagem Profunda, a qual permeia os fundamentos da Inteligência Artificial, é possível resolver tarefas mais complexas, por meio de redes neurais artificiais. À vista disso, este trabalho busca avaliar técnicas e algoritmos de Aprendizado de Máquina, mais precisamente Redes Neurais Convolucionais, no processo de monitoramento de tubarões, frente à identificação por imagens. Não obstante, configura-se como uma pesquisa aplicada, de caráter exploratório e com abordagem quantitativa, na qual foram avaliados os resultados obtidos pelas arquiteturas VGG 16, ResNet 50 e MobileNet V2, em meio ao processo de classificação. Por conseguinte, obtiveram-se resultados aceitáveis pelas métricas aplicadas, a exemplo do F1-Score, o qual alcançou valores próximos de 0.90. Portanto, tendo em vista o desempenho dos modelos, percebe-se a potencialidade desse experimento e sua importância para o processo de preservação, os quais podem ser explorados de forma mais abrangente, em cenários relacionados.

Palavras-chave: Monitoramento de tubarões; algoritmos de identificação de imagens; redes neurais convolucionais; aprendizagem de máquina.

ABSTRACT

In natural aquatic environments, bathers still experience shark attacks quite frequently. Thus, there is a clear need for more effective monitoring of these animals in order to ensure protection and also promote greater safety for citizens. Thus, through Deep Learning, which permeates the foundations of Artificial Intelligence, it is possible to solve more complex tasks using artificial neural networks. In view of this, this work seeks to evaluate Machine Learning techniques and algorithms, more precisely Convolutional Neural Networks, in the process of monitoring sharks, compared to image identification. Nevertheless, it is configured as an applied research, of an exploratory nature and with a quantitative approach, in which the results obtained by the VGG 16, ResNet 50 and MobileNet V2 architectures are evaluated, during the classification process. Therefore, we obtained acceptable results from the applied analyses, such as the F1-Score, which reached values close to 0.90. Therefore, considering the performance of the models, we can see the potential of this experiment and its importance for the preservation process, which can be explored more comprehensively in related scenarios.

Keywords: Shark monitoring; image identification algorithms; convolutional neural networks; machine learning.

LISTA DE SIGLAS

CEMIT	Comitê Estadual de Monitoramento de Incidentes com Tubarões
CNNs	Redes Neurais Convolucionais
AM	Aprendizagem de Máquina
NSW	New South Wales(Estado da Austrália)
R-CNN	Redes Neurais Convolucionais baseadas em regiões
UAV	Veículo Aéreo Não Tripulado
API	Interface de Programação de Aplicação
Colab	Google Colaboratory
VP	Verdadeiros Positivos
VN	Verdadeiros Negativos
FN	Falsos Negativos
FP	Falsos Positivos
GPU	Unidade de Processamento Gráfico
TPU	Unidade de Processamento de Tensor
ADAM	Estimativa de Momento Adaptável

SUMÁRIO

1. INTRODUÇÃO	1
2. REFERENCIAL TEÓRICO	3
2.1. Aprendizagem de Máquina	3
2.2. Redes Neurais Convolucionais	4
2.2.1. Camada de Convolução	6
2.2.2. Camada de Pooling	7
2.2.3. Camada totalmente conectada	9
3. TRABALHOS RELACIONADOS	10
4. AIMT - AVALIAÇÃO DE ALGORITMOS DE IDENTIFICAÇÃO DE IMAGENS PARA MONITORAMENTO DE TUBARÕES	13
4.1. Base de dados	13
4.2. Avaliação das arquiteturas	15
4.2.1. VGG 16	16
4.2.2. ResNet 50	16
4.2.3. MobileNet V2	16
4.3. Métricas de avaliação	18
5. EXPERIMENTO E RESULTADOS	19
6. CONCLUSÃO E TRABALHOS FUTUROS	24
REFERÊNCIAS	25

1. INTRODUÇÃO

É notório que os tubarões configuram-se como seres essenciais do ecossistema marinho, ao mesmo tempo em que ressaltam-se as dificuldades de equilibrar a preservação dessas espécies com a proteção das pessoas. Dessa forma, mediante uma análise multidimensional, percebe-se a complexidade das relações entre humanos e tubarões, o que sugere a necessidade de adotar estratégias mais abrangentes, para lidar com os ataques, tendo em vista tanto a conservação ambiental, quanto os aspectos sociais e culturais envolvidos (Silva e Nascimento, 2020).

Desse modo, diante do estado de Pernambuco, mais especificamente a cidade do Recife e região metropolitana, tem-se obtido por décadas registros frequentes de ataques de tubarões a banhistas e/ou surfistas, haja vista o reconhecimento mundial das belezas naturais, e assim, a atração de muitos turistas. Diante disso, é ponderável que a quantidade de pessoas nas praias, seja para banho ou para esportes, é apontada como uma das principais causas dos ataques. Entretanto, algumas ações como: a poluição dos rios, o processo de urbanização das praias, a implementação de complexos portuários (a exemplo de Suape), o aterramento de manguezais e o fluxo de navios podem causar impactos ambientais e atrair determinadas espécies (Silva, Nascimento, 2021).

Dessa forma, em meio a esses incidentes, é relutante o processo de monitoramento dos tubarões, introspectivo a medidas preventivas, para minimizar os riscos, seja diante da defesa civil, seja frente à conservação das espécies. Não obstante, tem-se a atuação do Comitê Estadual de Monitoramento de Incidentes com Tubarões (CEMIT), o qual integra diferentes entidades, a exemplo de pesquisadores, órgãos de segurança pública e instituições ambientais como membros. Assim, esse comitê deve atuar como centro de referência, mediante o registro e o acompanhamento dos casos, e da definição de estratégias e ações para redução dos incidentes (Silva, Nascimento, 2020; CEMIT, 2014).

Todavia, são evidentes os desafios frente ao monitoramento dos tubarões em seus habitats naturais, haja vista fatores como a rapidez em meio ao movimento, às condições variáveis da água, a qualidade das imagens capturadas (a exemplo da condição de iluminação), a presença de interferências como ondas e sombras na água e até mesmo o tamanho das espécies (tendo em vista diferentes ângulos de visão) (Purcell et al., 2022; Sharma et al., 2018). Desse modo, as Redes Neurais Convolucionais (CNNs) apresentam-se como potenciais para identificação de tubarões, por meio de registros visuais, em consonância com os padrões e/ou características das espécies, a fim de executar a tarefa de classificação (Purcell et al., 2022; Sharma et al., 2018; Gorkin III et al., 2020).

Destarte, o objetivo geral deste trabalho é avaliar técnicas e algoritmos de Aprendizado de Máquina, mais precisamente Redes Neurais Convolucionais, no processo de identificação de tubarões por imagens. À vista disso, será imprescindível a execução de propósitos específicos como: a realização de um estudo diante do estado da arte, em meio a soluções para possíveis ataques; a obtenção de uma base de dados ou capturas de imagens para realização de testes; e a aplicação dos procedimentos técnicos para análise e reconhecimento de imagens.

Por conseguinte, a viabilidade do estudo interpõe-se diante de uma melhor performance de rastreamento, frente à proteção e conservação dos tubarões, aliada à coleção de dados sobre as populações e os comportamentos, e possíveis respostas a mudanças em seus habitats (Purcell et al., 2022). Assim, evidenciam-se os benefícios em meio ao gerenciamento de áreas costeiras e a segurança dos banhistas (Sharma et al., 2018; Gorkin III et al., 2020).

Portanto, o trabalho possui a seguinte estrutura: a Seção 2 destina-se à fundamentação do estudo, em meio aos tópicos de Aprendizagem de Máquina e Redes Neurais Convolucionais. À vista disso, a Seção 3 apresenta os trabalhos relacionados, mediante uma comparação com o estudo em questão. A Seção 4, por sua vez, busca registrar os elementos e as técnicas aplicadas em meio ao processo de avaliação empregado. Assim, o tópico AIMT aborda a Base de Dados, a Avaliação das arquiteturas e as Métricas de avaliação. Ademais, a Seção 5 pondera características do experimento executado e exibe o

desempenho das redes, mediante os valores das métricas. Não obstante, a Seção 6 discute as conclusões a respeito do projeto e perspectivas para trabalhos futuros.

2. REFERENCIAL TEÓRICO

Esta Seção destina-se à apresentação da fundamentação teórica e dos trabalhos relacionados à temática, assim, divide-se em Seção 2.1 referente à Aprendizagem de Máquina e a Seção 2.2, onde são abordados os pontos sobre Redes Neurais Convolucionais e seus subtópicos.

2.1. Aprendizagem de Máquina

Inicialmente é essencial compreender o conceito de Aprendizagem de Máquina (AM), para contextualizar suas potenciais aplicações. Desse modo, pode-se defini-lo, diante do trabalho de (Mahesh, 2020), como o exame sistemático de algoritmos e modelos estatísticos, utilizados por sistemas de computador, para executar tarefas específicas sem programação direta. Essa explicação ressalta que a AM permite que os sistemas adquiram conhecimento dos dados e melhorem seu desempenho progressivamente.

Em conformidade, de acordo com Janiesch et al., (2021), a Aprendizagem de Máquina pode ser definida como a capacidade dos sistemas de adquirir conhecimento, a partir de dados de treinamento, adaptados a um problema específico, a fim de agilizar a construção de modelos analíticos e abordar tarefas relacionadas. Assim, a AM se esforça para deduzir de forma autônoma conexões e estruturas significativas, a partir de instâncias e percepções, em vez de depender de diretrizes explicitamente codificadas. Dessa forma, tem-se maior eficácia em diversas tarefas, à medida que o sistema acumula experiência com os dados.

Portanto, segundo (Mahesh, 2020; Janiesch et al., 2021; Sharma et al., 2020; Ludermir, 2021), é ponderável a projeção do Aprendizado de Máquina, diante de problemas de Classificação (no qual o objetivo é classificar os dados em categorias ou classes pré determinadas); Regressão (frente ao propósito de antecipar um resultado contínuo utilizando as informações de entrada

fornecidas); Clusterização (que tem a intenção de categorizar os dados em clusters ou grupos de acordo com características similares, sem depender de rótulos predeterminados); Detecção de Anomalias (na qual a intenção é reconhecer padrões que se desviam da norma prevista, sugerindo a presença de anomalias ou ocorrências pouco frequentes); e Aprendizado por Reforço (no qual o escopo configura-se mediante um agente, que adquire a capacidade de determinar decisões e comportamentos ideais, com o objetivo de maximizar a recompensa, em um ambiente em constante mudança.).

À vista disso, em meio a análise dos trabalhos de (Mahesh, 2020; Janiesch et al., 2021; Ludermir, 2021), percebe-se o potencial de aplicação da AM em eixos como: Mineração de Dados, Processamento de Vídeos e Imagens, Análises Preditivas, Reconhecimento de Padrões, Recomendações Personalizadas, Diagnósticos Médicos, Detecção de Fraudes, Análises de Sentimentos e Controle de Processos. Dessa forma, é notório a ampla variedade de aplicações da área, em vários domínios, o qual demonstra sua crescente influência e importância.

Contudo, diante da perspectiva de um conjunto maior ao qual se configura, a área de AM possui sub categorias fundamentais, em projetos mais específicos e complexos. Desse modo, em conformidade com a temática, a Aprendizagem Profunda é definida por (Janiesch et al., 2021; Sharma et al., 2020) como uma técnica de AM, que utiliza redes neurais artificiais, com várias camadas (ditas profundas), para modelar e resolver problemas mais elaborados.

2.2. Redes Neurais Convolucionais

Não obstante, em meio ao contexto de redes neurais artificiais, é possível evidenciar a aplicação das Redes Neurais Convolucionais (CNNs), diante de dados em formato não estruturado. Em conformidade, (Aloysius & Geetha, 2020; Li et al., 2018) estabelecem a introdução das CNNs para processar e interpretar dados organizados em formato de grade, como imagens e vídeos. Assim, elas apresentam ponderações substanciais como: Arquitetura Especializada, Camadas de Convolução, Camadas de Pooling e Hierarquia de Características ou Representações.

Todavia, as Redes Neurais Convolucionais sofreram modificações e ajustes ao longo do tempo, haja vista as demandas de aplicações e potenciais de inovação. Desse modo, é notório a importância de alguns fatos históricos, frente à implementação e o desenvolvimento das CNNs.

À vista disso, em meio aos trabalhos de (Aloysius & Geetha, 2020; Gu et al., 2018), evidenciam-se pontos como:

- **LeNet-5:** Uma das pioneiras entre as redes neurais convolucionais profundas, a qual empregou camadas convolucionais e de pooling para extrair características e reduzir a dimensionalidade dos dados. Assim, mostrou-se eficaz no reconhecimento de dígitos manuscritos, com base para a introdução de CNNs, em atividades de reconhecimento de padrões.

- **AlexNet:** A qual ganhou a competição anual de desempenho de algoritmos de reconhecimento de imagem (ImageNet), estabelecida como uma arquitetura com camadas profundas e métodos de regularização, como o dropout, com bons resultados em projetos de reconhecimento de imagens, de larga escala.

- **VGGNET:** Com uma estrutura que destacou-se por empregar camadas convolucionais pequenas (3x3) e empilhadas, o que aprimorou a profundidade e a capacidade de modelagem da rede. Além disso, utilizou camadas de pooling com dimensões de 2x2 pixels, as quais eram aplicadas após as duas convolucionais. Dessa forma, esse procedimento auxilia na redução da dimensionalidade das características extraídas e na diminuição do custo computacional.

- **GoogLeNet:** Uma inovação da época, que introduziu o conceito de Inception Modules, que possibilitou a criação de redes profundas e eficientes, ao integrar filtros de tamanhos variados, em uma única camada. Logo, era possível que a rede extraísse uma grande quantidade de características, em diferentes escalas e níveis de abstração.

- **ResNet:** O qual introduziu conexões de atalho (skip connections), com a permissão de que a informação e os gradientes se desloquem diretamente pela rede, e assim, contornam uma ou mais camadas intermediárias. Ademais, possibilitou o treinamento de redes extremamente profundas, com soluções para

os problemas de degradação de desempenho, que ocorrem devido a profundidade.

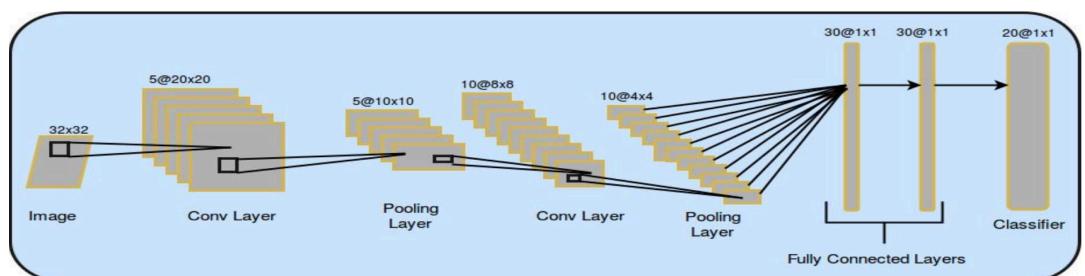
Portanto, as CNNs têm evoluído consideravelmente e têm sido amplamente aplicadas em vários campos, como visão computacional, em reconhecimento de objetos e segmentação de imagens, por exemplo, o qual demonstra um impacto relevante tanto em práticas industriais quanto científicas.

Diante disso, é fundamental a compreensão da arquitetura básica das Redes Neurais Convolucionais, que por sua vez são constituídas por camadas de convolução e de pooling, de maneira alternada, e por camadas totalmente conectadas, conforme apresentado na Figura 1, na Subseção 2.2.1, a seguir .

2.2.1. Camada de Convolução

Conforme (Li et al., 2018), a camada de convolução tem a função de utilizar filtros (ou kernels), para capturar características locais. Desse modo, esses filtros se movem sobre a entrada, para executar operações de convolução, que por sua vez geram mapas de características, os quais capturam informações importantes, como bordas, texturas e padrões.

Figura 1 - Arquitetura básica de ConvNet mostrando camadas alternadas de convolução e pooling. As pequenas caixas destacadas são as regiões receptivas. As conexões mostram a aprendizagem hierárquica implícita de características.



Fonte: (Aloysius & Geetha, 2020)

Além disso, (Neubauer, C., 1998) enfatiza que as camadas de convolução são desenvolvidas para identificar essas características de maneira hierárquica,

possibilitando que a rede aprenda representações progressivamente mais complexas da imagem, conforme avança pelo fluxo. Ademais, (Gu et al., 2018) complementa com a ponderação de que a aplicação de filtros com diferentes tamanhos e profundidades ajuda a aprimorar a eficácia na extração de características, em diversas escalas.

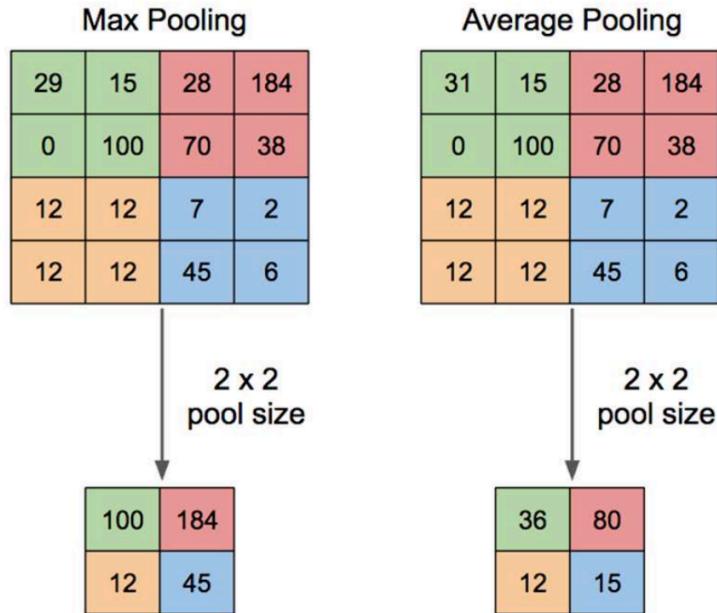
Não obstante, têm-se os parâmetros da camada de convolução, os quais são elencados pelos autores (Li et al., 2018; Neubauer, C., 1998; Gu et al., 2018) como:

- **Stride(Passo):** Determina o deslocamento dos filtros sobre a entrada. Assim, um stride maior tende a diminuir o tamanho dos mapas de características, enquanto um stride menor preserva mais informações.
- **Padding:** Determina a adição de margens à entrada, para manter a resolução e assegurar que as características nas bordas da imagem sejam também detectadas. À vista disso, (Gu et al., 2018) ressalta a substancialidade dele, para manter a dimensão espacial dos dados, por meio das camadas convolucionais.
- **Número de Filtros:** A quantidade de filtros em uma camada de convolução impacta na variedade de características que podem ser extraídas. Desse modo, caso apresentem um valor maior, possibilitam a captura de forma mais ampla.

2.2.2. Camada de Pooling

Diante da camada de pooling, (Li et al., 2018; Neubauer, C., 1998; Gu et al., 2018) ponderam a funcionalidade frente ao desenvolvimento, e consequentemente a diminuição da dimensionalidade dos mapas de características, gerados pela camada de convolução. Dessa forma, é possível reduzir a complexidade computacional e temporal de processamento, além de atuar como um mecanismo de regularização para a rede.

Figura 2 - Ilustração de Max Pooling e Average Pooling



Fonte: (Muhamad Yani et al, 2019)

Quanto aos tipos de operação, os autores destacam o Max Pooling e o Average Pooling:

- **Max Pooling:** Configura-se mediante a escolha do valor máximo em cada área da janela de pooling. Dessa forma, esse método é eficiente para destacar as características mais significativas e diminuir a dimensionalidade, a fim de manter as informações essenciais.
- **Average Pooling:** Interpõe a média dos valores dentro de cada área da janela de pooling, o qual fornece uma abordagem mais suave em relação ao max pooling.

Desse modo, em relação ao tamanho da janela de pooling, os autores já mencionados evidenciam o impacto direto em meio ao nível de redução das dimensões. Além disso, tem-se o Stride (passo), diante do deslocamento da janela, sobre o mapa de características. Assim, um stride maior tende a provocar uma redução mais significativa na dimensionalidade, enquanto um stride menor preserva mais detalhes na representação compactada.

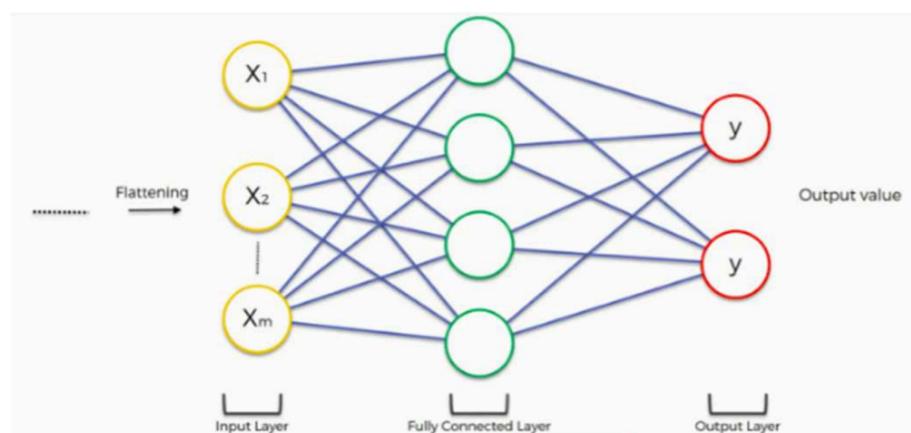
A vista disso, também é destaque dos autores, a interposição da camada de pooling, frente à invariância das características encontradas, em relação a variações de escala e posição. Logo, a rede se torna mais resistente a pequenas

variações, na posição ou na escala dos objetos nas imagens, o que garante maior eficiência e robustez às CNNs.

2.2.3. Camada totalmente conectada

Mediante o fluxo de processo, a camada totalmente conectada vincula-se à etapa final de saída da rede. Desse modo, subsequente a extração de características pelas camadas convolucionais e de pooling, tem-se a execução da função de classificação ou regressão, com base nas referências que foram previamente extraídas. Tal fato é ponderado por (Li et al., 2018; Neubauer, C., 1998; Gu et al., 2018; Aloysius & Geetha, 2020).

Figura 3 - Camada Oculta



Fonte: (Muhamad Yani et al, 2019)

Diante disso, ainda conforme os autores, nesse setor, cada neurônio se conecta aos da camada anterior, o que possibilita à rede integrar todas as características extraídas (combinação), para gerar uma decisão final sobre a classificação ou outro tipo de resultado. Portanto, tem-se uma contribuição para a redução da dimensionalidade dos dados, ao agrupar as informações obtidas das camadas anteriores em um vetor de características, que é utilizado para a decisão final.

Dessa forma, com as informações combinadas, realiza-se a interpretação, diante da última etapa, e com isso pode-se executar tarefas de classificação (em meio a entradas de classes já definidas) e previsão.

3. TRABALHOS RELACIONADOS

Em meio as ponderações realizadas, percebe-se a potencialidade de aplicação das Redes Neurais Convolucionais em projetos com atividades de detecção e classificação de objetos. Portanto, frente aos avanços derivados desses modelos, tem-se impulsionado as iniciativas de monitoramento aquático, diante das implementações. Dessa forma, têm-se a seguir uma tabela de comparação, com os principais pontos dos estudos na área, e as diferenças em evidência, em relação ao trabalho desenvolvido.

Tabela 1 - Propostas dos trabalhos relacionados e diferenças em relação ao estudo desenvolvido.

Autor(es)	Proposta do trabalho relacionado	Diferenças em relação ao estudo desenvolvido
Purcell et al., 2022	Examinar a eficácia de algoritmos comuns de deep learning, na detecção de criaturas marinhas, em vídeos capturados por drones, com ênfase na diferenciação entre espécies de tubarões.	A classificação dos animais marinhos, frente à quantidade de classes; o formato da base de dados utilizada; e a quantidade e versão das arquiteturas aplicadas.
Sharma et al., 2018	Explorar a eficácia das Redes Neurais Convolucionais baseadas em regiões (R-CNN), na detecção de diversos objetos marinhos, com	O número de classes/categorias a serem identificadas; a estrutura dos dados avaliados; e as redes empregadas.

	um foco particular em tubarões.	
Gorkin III et al., 2020	Descrever a plataforma Sharkeye, um projeto pioneiro para demonstrar alertas pessoais de tubarões para banhistas na água e em terra, aproveitando a coleta inovadora de imagens por UAV, algoritmos de detecção de aprendizado de máquina hospedados na nuvem e relatórios via dispositivos inteligentes.	A forma de desenvolvimento, diante do objetivo do trabalho.
Panico et al., 2020	Apresentar um novo algoritmo de detecção e rastreamento de objetos, que pode ser utilizado em placas de processamento de baixo consumo de energia. Em particular, analisar um cenário de aplicação específico no qual um UAV patrulha as linhas costeiras e classifica autonomamente diferentes tipos de objetos marinhos.	A quantidade de classes em meio aos objetos marinhos; o número de imagens colecionadas para a base de dados; as arquiteturas avaliadas.

Fonte: (Elaborada pelos autores)

No trabalho de (Purcell et al. 2022), utilizou-se um extenso banco de vídeos, os quais foram reunidos durante os Testes de Drones do Departamento de Indústrias Primárias de NSW desde 2016. Assim, os dados foram treinados por duas redes neurais, baseadas nas arquiteturas ResNet-50 e MobileNet V1, com o objetivo de identificar e classificar dez tipos de objetos marinhos, em vídeos com resolução de 1080p. Desse modo, percebe-se certas similaridades

em relação ao objetivo dos trabalhos comparados, contudo, são evidentes pontos de divergência entre eles. No caso deste estudo, o número de classes a serem detectadas são duas. Além disso, foi utilizada uma base de dados com imagens, com dimensões iniciais de 640x640 pixels, para avaliação das arquiteturas VGG 16, ResNet 50 e MobileNet V2.

Não obstante, em (Sharma et al., 2018), também é perceptível uma tangência em relação ao objetivo principal do trabalho em desenvolvimento, entretanto, há distinções diante das arquiteturas de rede aplicadas, a saber, Zeiler e Fergus (ZF), Visual Geometry Group (VGG16) e VGG_M, a fim de avaliar seu desempenho. Desse modo, os experimentos conduzidos no trabalho de Sharma et al utilizaram um conjunto de dados composto por 3957 quadros de vídeo, em comparação com o deste estudo, o qual teve como base uma coleção de imagens, com um total de 6953 registros.

Ademais, no estudo de (Gorkin III et al, 2020) foi apresentada a plataforma Sharkeye, uma inovação projetada para fornecer alertas pessoais sobre a presença de tubarões para banhistas, tanto na água quanto na terra. Assim, o sistema utiliza imagens aéreas captadas por drones, algoritmos de detecção baseados em aprendizado de máquina na nuvem e alertas enviados a dispositivos inteligentes. Diante disso, é perceptível uma diferença substancial em função dos objetivos dos trabalhos comparados, haja vista que esse envolveu a criação de um novo algoritmo de detecção, treinado com imagens reais de tubarões e arraias, capturadas em praias locais. Outrossim, o fato de que esse algoritmo foi implementado na nuvem e integrado com um aplicativo para smartwatches, que emite alertas automáticos.

Já em (Panico et al., 2020) percebe-se uma similaridade ainda maior em relação a estrutura da base de dados, haja vista que foi criado um banco com mais de 115.000 imagens para o treinamento e teste de várias arquiteturas de Redes Neurais Convolucionais (CNN). Além disso, foi introduzido um algoritmo adaptativo de detecção e rastreamento, que melhora a eficiência do sistema, mediante a otimização do equilíbrio entre a detecção de novos objetos e o rastreamento de alvos já existentes. Logo, suas diferenças em relação a este estudo remetem-se às arquiteturas a serem avaliadas, às categorias a serem identificadas e à quantidade de registros obtidos para análise.

Destarte, mediante à uma análise dos estudos anteriores, é pertinente destacar a proeminência e a exequibilidade da aplicação de CNNs em projetos aquáticos. Dessa forma, esse estudo fundamenta-se diante da aplicação de técnicas e algoritmos de Aprendizado de Máquina, no processo de monitoramento de tubarões, mediante a identificação por imagens.

4. AIMT - AVALIAÇÃO DE ALGORITMOS DE IDENTIFICAÇÃO DE IMAGENS PARA MONITORAMENTO DE TUBARÕES

O trabalho consiste em uma pesquisa aplicada, de caráter exploratório e com abordagem quantitativa. Assim, busca-se avaliar os resultados obtidos pelas arquiteturas VGG 16, ResNet 50 e MobileNet V2, mediante o processo de classificação dos tubarões, em meio às imagens, haja vista a divisão dos dados em treinamento, validação e teste. À vista disso, serão pontuados a seguir informações da base de dados utilizada, detalhes sobre o processo de avaliação aplicado e as métricas utilizadas no estudo.

4.1. Base de dados

A base de dados foi obtida no site Roboflow Universe e configura-se como um conjunto de imagens aéreas, em um total de 6953 registros, referente às classes: tubarão, bote, pessoa e outro animal (SHARKSPOTTING, 2024). Diante disso, é importante ressaltar a escolha desta coleção, frente à dificuldade em localizar uma base de dados consistente e adequada, em meio aos trabalhos colecionados e entre outros, haja vista que são disponibilizados registros e análises do desempenho das arquiteturas aplicadas, mas não das fontes consultadas.

Não obstante, foi necessário realizar ajustes em meio a organização das imagens obtidas, diante das pastas e do processo de renomeação. Assim, após as modificações, têm-se armazenados os dados em uma pasta de maior nível, com o nome de sharkdatabase, e esta por sua vez, possui os registros colecionados em três subpastas denominadas train, valid e test, referente à divisão dos conjuntos. Ademais, cada subpasta relativa aos conjuntos possui outras duas subpastas, associadas às classes, as quais foram definidas como

shark e no_shark. Dessa forma, as antigas classes - bote, pessoa e outro animal -, foram consideradas como uma única classe, no caso, no_shark ou não tubarão.

Figura 4 - Imagens da base de dados



Fonte: (Elaborada pelos autores)

Portanto, os conjuntos de treinamento, validação e teste possuem uma quantidade de imagens de 4172, 1391 e 1390 respectivamente. Assim, foi destinado 60% dos dados para treinamento, a fim de que o modelo possa aprender diante dos padrões; 20% para validação, como forma de analisar o desempenho dos modelos durante o treino, e assim, dar suporte na seleção dos hiperparâmetros, evitando *overfitting*; e 20% para teste, para avaliar o real desempenho das arquiteturas, com registros desconhecidos previamente.

Ademais, é fundamental que os conjuntos tenham imagens distintas, para que os rótulos não sejam memorizados e dessa forma, tenha-se uma diversidade e representatividade das classes, a fim de garantir uma avaliação mais justa e imparcial. Portanto, como não há informações específicas sobre como as imagens foram obtidas e não há registros de *augmentation* ou aumento dos dados, processos como o de zoom, flip horizontal, preenchimento das bordas e ajuste do brilho foram aplicados de forma adequada, aliado ao

redimensionamento delas, haja vista sua proporção inicial de 640x640 pixels e o tamanho esperado pelas arquiteturas de 224x224 pixels.

4.2. Avaliação das arquiteturas

Antes de ponderar as arquiteturas selecionadas para o trabalho, é substancial destacar as principais bibliotecas empregadas, a exemplo do Tensorflow, da API do Keras e do Scikit-learn. Além disso, foi utilizada a linguagem Python para dar suporte ao processo de avaliação, em razão da sua compatibilidade com as bibliotecas escolhidas e até mesmo com outras relacionadas.

Diante da aplicação de redes neurais profundas, da diversidade de algoritmos e parâmetros possíveis, foi adotado o Tensorflow. Em conformidade, ele configura-se como uma biblioteca de código aberto, que disponibiliza recursos avançados para implantar seus modelos em qualquer ambiente, a exemplo de servidores, dispositivos de borda, navegadores, dispositivos móveis, microcontroladores, entre outros (TensorFlow, n.d.). Assim, ele promove a simplificação, frente aos processos de criação, treinamento e até exportação de modelos, além de possuir interação com o Keras e com o Google Colaboratory, ou Colab.

Não obstante, o Keras apresenta-se de forma relevante diante da estrutura do projeto, tendo em vista o suporte para as redes neurais pré-treinadas; a consistência e simplicidade (como característica da API) em conjunto com o Tensorflow por exemplo; a minimização das ações dos usuários e a sinalização de mensagens de erro claras e acionáveis; (Keras, n.d.).

Outrossim, foi adotada a biblioteca Scikit-learn, caracterizada por ser de código aberto, por apresentar compatibilidade com a linguagem de programação Python e por colecionar modelos aplicáveis na área de Aprendizagem de Máquina. Desse modo, ela foi substancial diante do cálculo e da avaliação das métricas selecionadas. Ademais, vale ressaltar sua versatilidade, ao considerar sua integração com outras bibliotecas, em Python, como Matplotlib, NumPy, Seaborn, entre outras, que também foram empregadas no desenvolvimento da avaliação das arquiteturas (Scikit-learn, n.d.).

Destarte, diante das características das bibliotecas utilizadas, é imprescindível enfatizar os principais pontos referentes às aplicações das arquiteturas. A seguir têm-se:

4.2.1. VGG 16

Tendo em vista a necessidade de extrair características detalhadas e complexas das imagens, a fim de distinguir tubarões de outras entidades e ruídos, foi empregada a arquitetura VGG 16. Por conseguinte, ela caracteriza-se por ser mais leve em comparação com a VGG 19, tendo em vista, a configuração de 13 camadas convolucionais e 3 camadas totalmente conectadas. Além disso, costuma utilizar filtros 3x3 com stride 1 e padding 1, aliados a funções de ativação ReLU. Dessa forma, ela destaca-se diante sua profundidade e eficácia em tarefas de reconhecimento, o que impulsiona a precisão e a robustez do modelo (Sharma et al., 2018).

4.2.2. ResNet 50

Diante dos desafios na identificação de tubarões e o rendimento frente ao desempenho no processo de classificação das imagens, foi aplicada a arquitetura ResNet 50, como uma potencial instância, mediante a construção de modelos mais profundos e eficazes. Assim, ela destaca-se por apresentar uma estrutura composta por cerca de 50 camadas profundas e baseada em blocos residuais. Ademais, essas conexões, também chamadas de *skip connections*, facilitam o processo de treinamento, diante da habilidade em gerenciar redes neurais profundas, de maneira mais eficaz, o que reduz potenciais problemas de degradação (Purcell et al., 2022; Ulloa et al., 2020).

4.2.3. MobileNet V2

Segundo (Purcell et al., 2022), a arquitetura MobileNet destaca-se por sua eficácia e leveza, principalmente em situações que exigem processamento em tempo real, em dispositivos com restrições de recursos. Dessa forma, em contraposição às conexões residuais tradicionais, as conexões invertidas

buscam adicionar mais canais antes de reduzi-los. Assim, ela torna-se substancial, frente à capacidade de realizar detecção de tubarões de maneira eficiente e rápida, até mesmo diante de limitações de hardware. A vista disso, será adotado o modelo V2 dessa arquitetura, o qual remove a função de ativação ReLU, após a última camada de convolução 1x1, a fim de formar o gargalo linear (*linear bottleneck*).

Em conformidade com os resultados obtidos pelos modelos, foi aplicado o processo de otimização de hiperparâmetros, o qual configura-se como um algoritmo de busca da melhor combinação de valores e consequentemente em uma melhor performance do modelo empregado. Desse modo, utilizou-se a biblioteca do Keras Tuner, mediante a abordagem do Hyperband, para realizar a busca e os ajustes com os melhores hiperparâmetros (KERAS TUNER, 2025). Vale ressaltar que houve adequação de certos valores escolhidos para o processo de avaliação dos dados, tendo em vista a quantidade de registros a serem processados, as arquiteturas aplicadas, o procedimento de otimização (o qual descarta rapidamente as configurações menos promissoras) e consequentemente a execução do código, frente a potenciais algoritmos custosos.

Tabela 2 - Hiperparâmetros ajustáveis para os três modelos e seus respectivos valores testados

Hiperparâmetros	Valores Testados
Quantidade de neurônios na primeira camada densa ou totalmente conectada	min_value=256 max_value=512 step=128
Quantidade de neurônios na segunda camada densa ou totalmente conectada	min_value=128 max_value=256 step=64
Taxa de aprendizado ou learning_rate	values=[1e-3, 1e-4, 1e-5]

Fonte: (Elaborada pelos autores)

4.3. Métricas de avaliação

Haja vista o tipo de tarefa processada pelos modelos, são necessárias medidas ou métricas de avaliação de desempenho, a fim de mensurar a precisão e a capacidade de generalização. Desse modo, elas são imprescindíveis, frente à medição da performance e da eficácia de algoritmos, em diferentes aplicações. Assim, deve-se escolher aquela a qual se adequa melhor ao problema ou até gerar a combinação de várias para proporcionar uma visão mais abrangente (Mahesh, 2020; Janiesch et al., 2021; Sharma et al., 2020).

A vista disso, por se tratar de uma tarefa de classificação, mais precisamente diante das classes tubarão e não-tubarão, serão adotadas as métricas de acurácia, matriz de confusão e F1-Score. Dessa forma, será feita uma análise criteriosa e abrangente, em meio aos padrões avaliados, seja mediante o resultado da relação de valores, como a acurácia e o f1-score, seja com base no detalhamento da distribuição das previsões, a exemplo da matriz de confusão.

Figura 5 - Matriz de confusão

	Predito Positivo	Predito Negativo
Real Positivo	Verdadeiros Positivos	Falsos Negativos
Real Negativo	Falsos Positivos	Verdadeiros Negativos

Descrição: Verdadeiros positivos (VP) e verdadeiros negativos (VN) indicam as previsões corretas, enquanto falsos negativos (FN) e falsos positivos (FP) indicam as previsões incorretas.

Fonte: (Elaborada pelos autores)

Figura 6 - Fórmula do F1 Score

$$F1\text{-Score} = \frac{2 * TP}{2 * TP + FP + FN}$$

Fonte: (Elaborada pelos autores)

Figura 7 - Fórmula da Acurácia

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Fonte: (Elaborada pelos autores)

5. EXPERIMENTO E RESULTADOS

Tendo em vista o objetivo geral do trabalho, de avaliar técnicas e algoritmos de Aprendizado de Máquina, mais precisamente Redes Neurais Convolucionais, no processo de monitoramento de tubarões, frente à identificação por imagens, esta seção apresenta a forma como o experimento foi conduzido, os resultados obtidos pelo desempenho das redes e suas avaliações em meio aos valores das métricas.

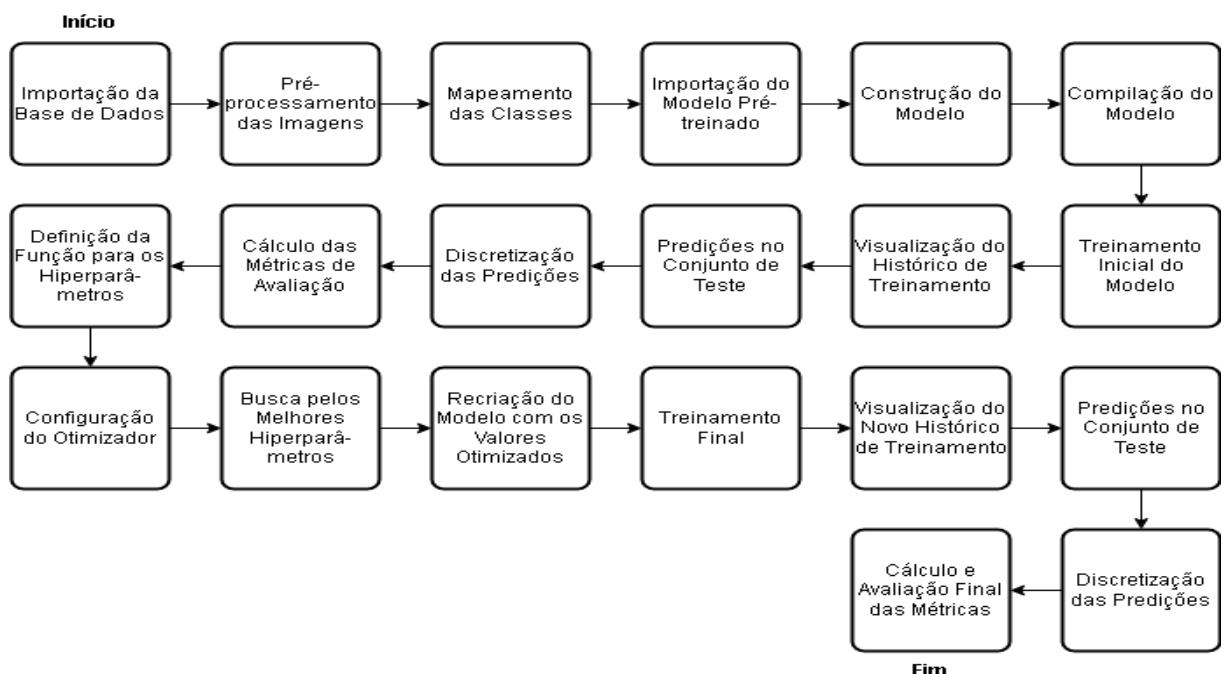
À vista disso, foi aplicado a linguagem Python, para a execução dos algoritmos no ambiente de desenvolvimento, aliado à utilização da plataforma Google Colaboratory ou Colab, diante da compatibilidade com as bibliotecas adotadas. Assim, ela configura-se como um ambiente de programação e de execução de notebooks Jupyter. Além disso, possui suporte para outros recursos como o acesso direto via navegadores web, hardwares de alto desempenho como GPUs e TPUs, colaboração em tempo real e integração com o Google Drive, o qual também foi utilizado no trabalho (Ferreira et al., 2023).

Diante das arquiteturas, foram realizados processos semelhantes em meio ao treinamento dos modelos, a fim de avaliar os resultados sem a otimização de hiperparâmetros e após sua aplicação, e com isso obter as melhores respostas, frente às métricas analisadas. Dessa forma, realizou-se o descongelamento de algumas camadas, respectivas aos modelos adotados e pré-treinados da base ImageNet (VGG 16, ResNet 50, MobileNet V2), para

ajustá-las ao conjunto de dados, mediante o treinamento, e assim potencializar o desempenho. Logo, busca-se um equilíbrio entre a reutilização de informações previamente aprendidas no ImageNet e a adaptação às novas coleções (Shermin et al., 2019).

Não obstante, foi utilizado o otimizador ADAM diante das redes neurais profundas, haja vista sua capacidade de adaptação e o ajuste da taxa de aprendizado, para cada parâmetro. Desse modo, tem-se uma combinação dos conceitos referentes a momentum e adaptabilidade, na qual é permissível a utilização de médias móveis exponenciais dos gradientes de primeira e segunda ordem, para atualizar os pesos da rede (Dereich & Jentzen, 2024; Reddi et al., 2018). À vista disso, realizou-se o processo de treinamento, durante 10 épocas, tendo em vista a robustez das arquiteturas escolhidas, a quantidade de registros colecionados e a execução dos modelos.

Figura 8 - Fluxograma das etapas aplicadas na construção do experimento.



Fonte: (Elaborada pelos autores)

Portanto, após o treinamento e a geração das previsões, foram obtidos os resultados das métricas aplicadas às redes no conjunto de teste. Assim, têm-se:

Tabela 3 - Resultados das métricas sem a otimização de hiperparâmetros

Redes/Métricas	Acurácia	F1-Score	Matriz de confusão
VGG 16	0,9676	0,9036	$\begin{bmatrix} 211 & 36 \\ 9 & 1134 \end{bmatrix}$
ResNet 50	0,9432	0,8323	$\begin{bmatrix} 196 & 51 \\ 28 & 1115 \end{bmatrix}$
MobileNet V2	0,9727	0,9195	$\begin{bmatrix} 217 & 30 \\ 8 & 1135 \end{bmatrix}$

Fonte: (Elaborada pelos autores)

A vista dos resultados, percebe-se um desempenho geral significativo das redes, com maior evidência para a arquitetura MobileNet V2, com o maior valor de acurácia (0,9727) e o melhor valor referente ao F1-Score (0,9195). Ademais, é perceptível uma taxa menor para os falsos positivos (8 FP) e os falsos negativos (30 FN), mediante a evidência da matriz de confusão, o que enfatiza a potencialidade de aplicação desse modelo para o processo de classificação, diante da identificação por imagens.

Outrossim, identificou-se um desempenho expressivo, por parte da VGG 16, e até relativamente próximo em comparação com a MobileNet V2. Em conformidade, apresenta uma acurácia de 0,9676 e o F1-Score de 0,9036. Desse modo, é considerável que as arquiteturas aplicadas demonstraram boa performance para a atividade em questão, além da possibilidade de substituição, a exemplo de contextos com restrições de recursos computacionais. Entretanto, a ResNet 50, embora configura-se como uma rede mais profunda e poderosa, gerou as métricas com os menores valores, em meio aos três modelos, para a acurácia e para o F1-Score.

Dessa forma, tem-se um indicativo de que a rede ResNet 50 não esteja capturando tão bem os padrões necessários para o processo de classificação. Além disso, apesar do desempenho considerável em relação à MobileNet V2 e a VGG 16, obteve-se uma quantidade razoavelmente expressiva de falsos negativos, os quais podem ser reduzidos mediante procedimentos de refinamento. À vista disso, percebe-se a importância de aplicar a otimização de hiperparâmetros, a fim de aprimorar os resultados, mediante uma melhor exploração das arquiteturas.

Por conseguinte, foi empregada a busca pelos melhores conjuntos de hiperparâmetros, mediante a utilização da biblioteca do Keras Tuner, para executar novamente o treinamento do modelo, com os valores mais adequados. Em consonância, foi aplicada uma técnica de parada antecipada ou *early stop*, com atribuição do valor 3 para o patience, a fim de prevenir o *overfitting* e melhorar o desempenho do modelo, o qual é sugerido pelos autores em (Hussein & Shareef, 2024).

Destarte, tem-se a seguir os resultados das métricas aplicadas às redes no conjunto de teste:

Tabela 4 - Resultados das métricas com a otimização de hiperparâmetros

Redes/Métricas	Acurácia	F1-Score	Matriz de confusão
VGG 16	0,9626	0,8839	[[198 49] [3 1140]]
ResNet 50	0,9525	0,8527	[[191 56] [10 1133]]
MobileNet V2	0,9727	0,9195	[[217 30] [8 1135]]

Fonte: (Elaborada pelos autores)

Diante disso, em comparação com os resultados sem a otimização de hiperparâmetros, percebe-se uma redução de falsos positivos, em meio a rede VGG 16, o que ressalta uma melhora na detecção da classe shark, e com isso garante melhoria na precisão e maior confiabilidade na classificação. Entretanto, a acurácia sofreu uma diminuição sutil e houve um certo aumento de falsos negativos, o que reduziu suavemente a precisão frente à classe no_shark.

Ademais, a arquitetura da ResNet 50 apresentou uma evolução em relação a acurácia (0,9525) e o F1-Score (0,8527). Além disso, foi observado uma redução significativa dos falsos positivos, o que indica uma melhor detecção para a classe shark, ao considerar os possíveis ajustes diante da otimização do modelo. Contudo, percebe-se um aumento discreto dos falsos negativos.

Todavia, a rede MobileNet V2 manteve os resultados expressivos, o que sugere um tangenciamento prévio frente à otimização ou a ausência de um impacto ainda mais significativo pelos hiperparâmetros ajustados. Dessa forma, torna-se evidente o bom desempenho em relação a estabilidade dessa arquitetura e o processo de classificação, em paralelo à atuação da VGG 16, a qual apresentou a menor quantidade de falsos positivos (apenas 3).

Portanto, é perceptível a potencialidade de aplicação dessas arquiteturas, na tarefa de classificação, mediante a identificação por imagens, no monitoramento de tubarões. Desse modo, pode-se destacar a importância do conjunto de dados para o desenvolvimento do trabalho, em razão da condução do experimento com registros reais, em meio ao ambiente aquático e a análise de desempenho diante das classes em questão.

Ademais, frente às capturas e os cenários apresentados, foi possível avaliar o comportamento dos modelos, em meio a fatores como às condições variáveis da água, a qualidade das imagens capturadas (a exemplo da condição de iluminação), a presença de interferências e até mesmo o tamanho das espécies (tendo em vista diferentes ângulos de visão). Dessa forma, recomenda-se para trabalhos futuros, a coleta de imagens com sensores de alta precisão, a fim de melhorar os registros a serem colecionados e a confiabilidade das redes.

6. CONCLUSÃO E TRABALHOS FUTUROS

Mediante o objetivo central do trabalho, de avaliar as arquiteturas de Redes Neurais Convolucionais, na identificação de tubarões por imagens, foi possível comparar os resultados obtidos pelas redes, por meio das métricas de Acurácia, F1-Score e Matriz de Confusão. Diante disso, obteve-se um desempenho geral significativo, com maior destaque para o modelo da MobileNet V2. Entretanto, em busca de refinamento em meio à tarefa de classificação, foi aplicada a otimização de hiperparâmetros, por meio do Keras Tuner.

Dessa forma, percebeu-se uma melhora em relação aos valores das métricas geradas pelas arquiteturas e uma estabilização, diante de um possível tangenciamento prévio de desempenho ou da ausência de um impacto significativo pelos hiperparâmetros ajustados. Assim, considerando que o experimento conta com uma coleção de dados reais, foi possível analisar o comportamento das redes, com base em cenários que apresentam condições variáveis referentes à água, à iluminação na captura das imagens, eventuais interferências e posicionamentos dos tubarões.

Portanto, por meio da coleção utilizada e dos resultados obtidos, os quais tangenciam a faixa de 90% em acurácia e F1-Score, por exemplo, percebe-se relevância no experimento desenvolvido. Além disso, mediante a capacidade de processamento disponível, em meio ao nível de recursos computacionais, aliado à quantidade de registros fornecidos, também é possível ressaltar o potencial impacto da aplicação dessas redes no monitoramento aquático.

Não obstante, tendo em vista o cumprimento do objetivo central do estudo, é notória a permeabilidade para processos de aprimoramento, em função dos resultados gerados, considerando, ainda, os elementos de suporte utilizados no trabalho. Desse modo, para o desenvolvimento de trabalhos futuros, é ponderável o compartilhamento de informações, mediante órgãos e responsáveis pela conservação dos tubarões, a fim de ampliar os cenários a serem monitorados e a aplicação de recursos computacionais mais robustos para a exploração de melhores configurações, o que possibilitaria a reavaliação do desempenho.

REFERÊNCIAS

- ALOYSIUS, N.; GEETHA, M. **A review on deep convolutional neural networks**, 2020.
- COMITÊ ESTADUAL DE MONITORAMENTO DE INCIDENTES COM TUBARÕES – CEMIT. **Regimento Interno**, 2014.
- DEREICH, S.; JENTZEN, A. **Convergence rates for the Adam optimizer**. arXiv preprint arXiv:2407.21078, 2024. Disponível em: <https://arxiv.org/abs/2407.21078>. Acesso em: 04 jun. 2025.
- FERREIRA, R. S.; CANESCHE, M.; PENHA, J. **Google Colab para Ensino de Computação**. Anais Estendidos do Simpósio Brasileiro de Educação em Computação, 2023.
- GORKIN III, R. et al. **Sharkeye: Real-Time Autonomous Personal Shark Alerting via Aerial Surveillance**, 2020.
- GU, J. et al. Recent advances in convolutional neural networks. **Pattern Recognition**, 2018.
- HUSSEIN, B. M.; SHAREEF, S. M. **An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning**, 2024.
- JANIESCH, C.; ZSCHECH, P.; HEINRICH, K. Machine learning and deep learning. **Electronic Markets**, 2021.
- KERAS. **Keras Documentation** [Documentação]. Disponível em: https://keras.io.translate.goog/?_x_tr_sl=en&_x_tr_tl=pt&_x_tr_hl=pt-BR&_x_tr_pto=sc. Acesso em: 04 jun. 2025.
- KERAS TUNER. **Keras Tuner**. Disponível em: https://keras.io/keras_tuner/. Acesso em: 04 jun. 2025.
- LI, Z. et al. **A survey of convolutional neural networks: Analysis, applications, and prospects**, 2018.
- LUDERMIR, T. B. **Inteligência Artificial e Aprendizado de Máquina: Estado Atual e Tendências**, 2021.
- MAHESH, B. **Machine learning algorithms – A review**, 2020.
- NEUBAUER, C. **Evaluation of Convolutional Neural Networks for Visual Recognition**, 1998.
- PANICO, A.; ZANOTTI FRAGONARA, L.; AL-RUBAYE, S. **Adaptive detection tracking system for autonomous UAV maritime patrolling**, 2020.

PURCELL, C. R. et al. **Assessing the ability of deep learning techniques to perform real-time identification of shark species in live streaming video from drones**, 2022.

REDDI, S. J.; KALE, S.; KUMAR, S. **On the convergence of Adam and beyond**. *International Conference on Learning Representations (ICLR)*, 2018.

ROBOFLOW. **Roboflow: Create and Deploy Computer Vision Models**. Disponível em: <https://roboflow.com>. Acesso em: 04 jun. 2025.

SCIKIT-LEARN. **Scikit-learn: Machine Learning in Python**. Disponível em: <https://scikit-learn.org/stable/>. Acesso em: 04 jun. 2025.

SHARKSPOTTING. **Sharkspotting Dataset**. Roboflow Universe, Roboflow, 2024. Disponível em: <https://universe.roboflow.com/sharkspotting-uwbou/sharkspotting-nixfq>. Acesso em: 04 jun. 2025.

SHERMIN, T. et al. **Enhanced Transfer Learning with ImageNet Trained Classification Layer**. arXiv preprint arXiv:1903.10150, 2019. Disponível em: <https://arxiv.org/abs/1903.10150>. Acesso em: 04 jun. 2025.

SHARMA, N.; SCULLY-POWER, P.; BLUMENSTEIN, M. **Shark detection from aerial imagery using region-based CNN, a study**, 2018.

SHARMA, N.; SHARMA, R.; JINDAL, N. Machine Learning and Deep Learning Applications – A Vision. **Journal of King Saud University - Computer and Information Sciences**, 2020.

SILVA, A. C. R.; NASCIMENTO, R. M. **Aprendendo a conviver com os tubarões: relações entre humanos e não humanos em Recife e no Arquipélago de Fernando de Noronha (BRA)**, 2021.

SILVA, A. C. R.; NASCIMENTO, R. M. **Ataques de tubarões: Relações multi espécie e gênero nas praias de Pernambuco, Brasil**, 2020.

TAM, A. W. T. et al. **Application of transfer learning using convolutional neural network method for early detection of Terry's nail**, 2019.

TENSORFLOW. **Learn TensorFlow**. Disponível em: <https://www.tensorflow.org/learn?hl=pt-br>. Acesso em: 04 jun. 2025.

ULLOA, G. et al. **Hammerhead shark detection using regions with convolutional neural networks**, 2020.