

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

DAIVES KAWON CHU

**Uma abordagem para detecção de serpentes
de interesse em saúde pública no Brasil por
meio de imagens**

Monografia apresentada como requisito parcial
para a obtenção do grau de Bacharel em Ciência
da Computação

Orientador: Prof. Dr. Dennis Giovani Balreira

Porto Alegre
2024

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos André Bulhões Mendes

Vice-Reitora: Prof^ª. Patricia Helena Lucas Pranke

Pró-Reitora de Graduação: Prof^ª. Cíntia Inês Boll

Diretora do Instituto de Informática: Prof^ª. Carla Maria Dal Sasso Freitas

Coordenador do Curso de Ciência de Computação: Prof. Marcelo Walter

Bibliotecário-chefe do Instituto de Informática: Alexsander Borges Ribeiro

“What about side by side with a friend?”

— LEGOLAS GREENLEAF

AGRADECIMENTOS

Primeiramente, gostaria de agradecer aos meus pais, Liu Yan Ling e David Cheuk Yam Chu, dos quais suporte e apoio foi extremamente fundamental durante toda a minha jornada acadêmica, sem eles nada disto seria possível.

Agradeço também aos meus amigos e amigas. Obrigado por estarem comigo nos momentos de celebrações e nos momentos de tensão, vocês fazem parte das minhas vitórias.

Gostaria de agradecer à minha companheira Julia Fröhlich por me apoiar durante o desenvolvimento deste projeto.

Por último, mas não menos importante, gostaria de agradecer ao meu orientador, Dennis Giovani Balreira, cujo apoio e orientações foram fundamentais para o sucesso deste projeto.

RESUMO

A detecção de animais peçonhentos é de extrema importância para a segurança pública e para a preservação da vida selvagem. O conhecimento antecipado da presença desses animais em determinadas áreas pode prevenir acidentes graves e até mesmo salvar vidas. No entanto, a identificação manual desses animais é uma tarefa desafiadora e arriscada, exigindo profissionais com formações específicas que nem sempre estão disponíveis em determinadas regiões. Este trabalho busca apresentar uma nova abordagem de detecção de serpentes de interesse em saúde pública no Brasil por meio de imagens, utilizando técnicas de aprendizado de máquina e apresentar um meio de aquisição de imagens de serpentes da fauna brasileira para estudos futuros. Para isso, criamos o conjunto de dados e criamos modelos de aprendizado de máquina baseados em quatro arquiteturas ANN com um conjunto de dados contendo 9236 imagens desses animais. Dentre as arquiteturas testadas, VGG, ResNet, DenseNet e Inception V3, a arquitetura que apresentou melhor resultado foi a VGG, conseguindo atingir uma acurácia de 85,9% e um *F1-score* de 63,5%. Os resultados atingidos tem o potencial de auxiliar na prevenção de acidentes e contribuir para a preservação da vida tanto dos seres humanos quanto dos próprios animais, além de fornecer uma ferramenta para profissionais de saúde, pesquisadores e socorristas que atuam na área da saúde e segurança em todo o país.

Palavras-chave: Inteligência Artificial. Serpentes Peçonhentas. Classificação de Serpentes.

An approach to the detection of public health-relevant snakes in Brazil through images

ABSTRACT

The detection of venomous animals is extremely important for public safety and wildlife preservation. Advanced knowledge of the presence of these animals in certain areas can prevent serious accidents and even save lives. However, manual identification of these animals is a challenging and risky task, requiring professionals with specific training not always available in certain regions. This work seeks to present a new approach to detecting snakes of public health interest in Brazil through images, using machine learning techniques and providing a means of acquiring images of snakes from Brazilian fauna for future studies. For this, we created a dataset and developed machine learning models based on four ANN architectures with a dataset containing 9,236 images of these animals. Among the tested architectures, VGG, ResNet, DenseNet, and Inception V3, VGG showed the best performance, achieving an accuracy of 85.9% and an F1-score of 63.5%. The achieved results have the potential to assist in accident prevention and contribute to the preservation of the life of both humans and the animals themselves, in addition to providing a tool for health professionals, researchers, and first responders working in health and safety across the country.

Keywords: Artificial Intelligence. Venomous Snakes. Snake Classification.

LISTA DE FIGURAS

Figura 1.1	Série histórica de acidentes causados por serpentes, aranhas e escorpiões notificados no SINAN, Brasil, 2012 a 2021	11
Figura 1.2	Relação entre número de acidentes, tempo entre acidente e atendimento e taxa de letalidade, Brasil, 2021 (N = 31.354).	12
Figura 2.1	Representação esquemática de um modelo simplificado de neurônio artificial.	16
Figura 2.2	Representação esquemática de uma Rede Neural Artificial multicamadas, mostrando a camada de entrada com dois atributos, múltiplas camadas intermediárias de neurônios e uma camada de saída, todas interligadas por conexões.	16
Figura 2.3	Representação esquemática de uma CNN, mostrando a camada convolutiva para detectar características, as camadas de <i>pooling</i> e as camadas finais totalmente conectadas.	18
Figura 2.4	Esquema de um bloco residual com duas camadas e a conexão de atalho.	18
Figura 2.5	Esquema de uma DenseNet com três blocos densos.	19
Figura 2.6	Figura ilustrativa de uma matriz de confusão para um problema de classificação binária.	21
Figura 2.7	Figura ilustrativa de um <i>3-fold cross-validation</i> , evidenciando os subconjuntos de treino e de teste.	23
Figura 3.1	Exemplo de imagens no <i>dataset Snakes of Perlis Corpus</i>	26
Figura 3.2	Exemplo de imagens no <i>dataset</i> usado por Dube et al. (DUBE; BHURU, 2022).	27
Figura 3.3	Exemplo de serpentes <i>Russell's Viper</i> e <i>Indian Rock Python</i> , respectivamente, utilizadas no trabalho de Dandeniya et al. (DANDENIYA; WICKRAMASINGHE; DASANAYAKA, 2022).	28
Figura 3.4	Exemplo de serpentes <i>Malayopython reticulatus</i> e <i>Calloselasma rhodostoma</i> , respectivamente, utilizadas no trabalho de Othman et al. (OTHMAN et al., 2021).....	28
Figura 4.1	Fluxograma da metodologia do trabalho desde a obtenção e criação do conjunto de dados à criação e avaliação dos modelos.	30
Figura 4.2	Exemplos de imagens presentes no <i>dataset</i>	32
Figura 4.3	Exemplos de imagens pré-processadas.	33
Figura 5.1	Matriz de confusão da validação cruzada na VGG com <i>batch size</i> de 7500 e 35 épocas de treinamento.	37
Figura 5.2	Exemplos de corais presentes na fauna brasileira.	38
Figura A.1	<i>Batch size</i> de 256 e 20 épocas de treinamento.	45
Figura A.2	<i>Batch size</i> de 256 e 35 épocas de treinamento.	46
Figura A.3	<i>Batch size</i> de 512 e 20 épocas de treinamento.	47
Figura A.4	<i>Batch size</i> de 512 e 35 épocas de treinamento.	48
Figura A.5	<i>Batch size</i> de 7500 e 20 épocas de treinamento.	49
Figura A.6	<i>Batch size</i> de 7500 e 35 épocas de treinamento.	50

LISTA DE TABELAS

Tabela 5.1 Desempenho das arquiteturas com <i>batch size</i> de 256 e 20 épocas de treinamento.	35
Tabela 5.2 Desempenho das arquiteturas com <i>batch size</i> de 256 e 35 épocas de treinamento.	35
Tabela 5.3 Desempenho das arquiteturas com <i>batch size</i> de 512 e 20 épocas de treinamento.	36
Tabela 5.4 Desempenho das arquiteturas com <i>batch size</i> de 512 e 35 épocas de treinamento.	36
Tabela 5.5 Desempenho das arquiteturas com <i>batch size</i> de 7500 e 20 épocas de treinamento.	36
Tabela 5.6 Desempenho das arquiteturas com <i>batch size</i> de 7500 e 35 épocas de treinamento.	37

LISTA DE ABREVIATURAS E SIGLAS

ANN	<i>Artificial Neural Network</i>
CNN	<i>Convolutional Neural Network</i>
SINAN	Sistema de Informação de Agravos de Notificação
SAMU	Serviço de Atendimento Móvel de Urgência
CNES	Cadastro Nacional de Estabelecimentos de Saúde
VGG	<i>Visual Geometry Group</i>

SUMÁRIO

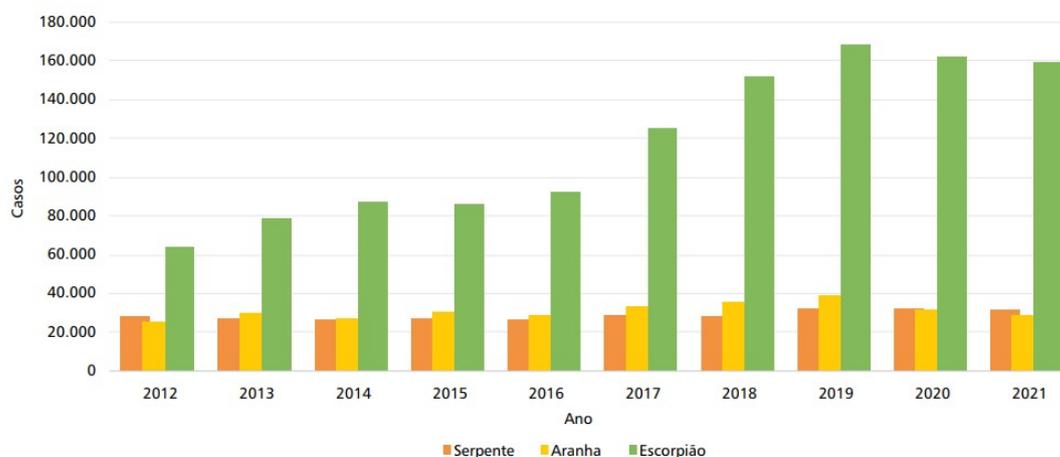
1 INTRODUÇÃO	11
1.1 Motivação	11
1.2 Objetivos	13
1.3 Organização	13
2 CONCEITOS BÁSICOS	14
2.1 Serpentes de interesse em saúde pública	14
2.2 Redes Neurais Artificiais	15
2.2.1 Rede Neural Convolucional	17
2.2.2 Arquiteturas.....	17
2.2.3 Transfer learning	20
2.2.4 Métricas de avaliação.....	20
2.2.5 Métodos de avaliação de modelos	22
2.2.6 Hiperparâmetros.....	23
3 TRABALHOS RELACIONADOS	25
4 METODOLOGIA	30
4.1 Dataset	30
4.1.1 Filtragem de Dados	31
4.2 Pré-processamento das imagens	31
4.3 Classificadores	33
5 RESULTADOS E DISCUSSÃO	35
6 CONCLUSÃO	39
6.1 Limitações	39
6.2 Trabalhos futuros	40
REFERÊNCIAS	41
APÊNDICE A — MATRIZES DE CONFUSÃO DAS ARQUITETURAS TESTADAS	45

1 INTRODUÇÃO

A prevenção e o tratamento de acidentes ofídicos é uma preocupação significativa para a nossa sociedade devido ao fato de que o Brasil é um dos países com maior biodiversidade de serpentes do mundo (MINISTÉRIO DA SAÚDE, 2022). Esses acidentes podem causar sequelas permanentes e serem potencialmente fatais, representando um sério problema de saúde pública no país, especialmente para populações de trabalhadores da agropecuária, comunidades ribeirinhas e povos originários.

Segundo um Boletim Epidemiológico da Secretaria de Vigilância em Saúde, no ano de 2021, acidentes por animais peçonhentos foram responsáveis por 257.073 registros no SINAN. Deste total de notificações, 31.354 foram acidentes ocasionados por serpentes (acidentes ofídicos), o que corresponde a 12,20% dos registros do SINAN, o segundo mais frequente dentre os tipos de acidente por animais peçonhentos, como pode ser visto na Figura 1.1.

Figura 1.1 – Série histórica de acidentes causados por serpentes, aranhas e escorpiões notificados no SINAN, Brasil, 2012 a 2021



Fonte: (MINISTÉRIO DA SAÚDE, 2022).

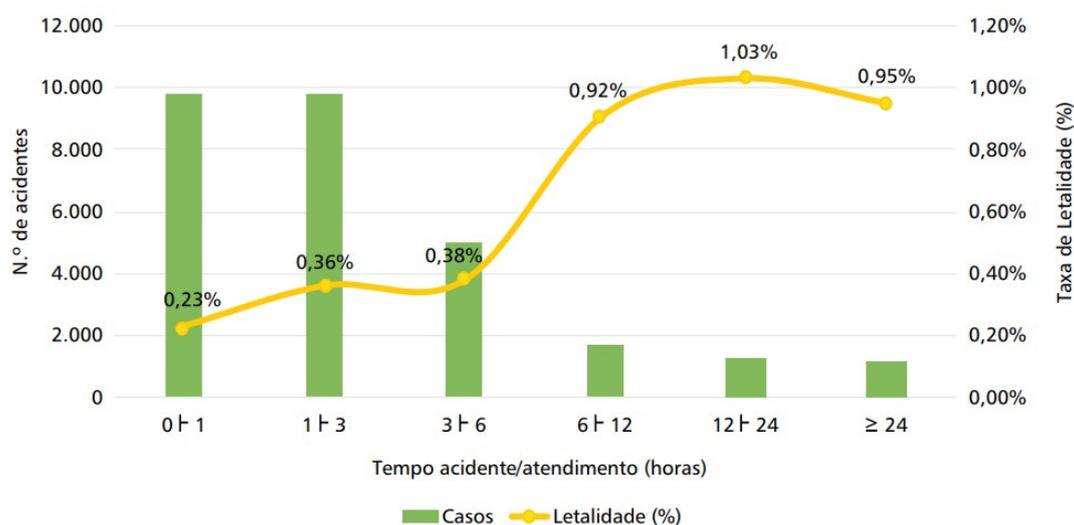
1.1 Motivação

Com o intuito de promover o acesso a atendimento em casos de acidentes ofídicos, que é o quadro clínico decorrente da mordedura de serpentes, o poder público disponibiliza informações sobre hospitais de referência especializados em soroterapia através de uma lista abrangente desses hospitais, incluindo detalhes como localização, nome, ende-

reço, telefone e o CNES. Além disso, há destaque também para contatos de emergência como o Serviço de Atendimento Móvel de Urgência (192), o Corpo de Bombeiros (193) e o Centro de Informação e Assistência Toxicológica, este último possuindo números de telefone específicos para cada estado e região.

Dados do SINAN mostram que a demora para o atendimento após um acidente ofídico é um dos fatores que aumentam a taxa de letalidade destes acidentes (MINISTÉRIO DA SAÚDE, 2022) ressaltado pela Figura 1.2.

Figura 1.2 – Relação entre número de acidentes, tempo entre acidente e atendimento e taxa de letalidade, Brasil, 2021 (N = 31.354).



Fonte: (MINISTÉRIO DA SAÚDE, 2022).

A detecção de serpentes de interesse em saúde pública no Brasil é importante para a implementação de estratégias eficazes de gestão de saúde pública. Ao determinar se uma serpente é de interesse em saúde pública, ou seja, se ela representa um risco significativo para a população devido ao seu potencial venenoso, é possível priorizar recursos e esforços de educação e prevenção em áreas onde essas espécies são mais prevalentes. Isso inclui a preparação adequada de serviços de emergência e a disponibilidade de anti-venenos específicos, bem como campanhas de conscientização pública sobre como evitar acidentes e proceder em caso de mordeduras. A rápida identificação dessas serpentes contribui diretamente para a redução do tempo de atendimento e, conseqüentemente, para a diminuição da taxa de letalidade em acidentes ofídicos.

O uso de redes neurais para a classificação de serpentes é um tema muito explorado recentemente como pode ser visto no artigo "Snake species classification using

deep learning techniques"(AHMED; GAD; ABOUTABL, 2023). Nele, os autores usam técnicas de aprendizado de máquina para classificar espécies de serpentes para fins de tratamento em caso de acidente ofídicos. Dube and Bhuru (2022) desenvolveram um aplicativo para classificar espécies de serpentes com auxílio de técnicas de redes neurais. Também vale citar um trabalho publicado em 2021 de Progga et al. (2021), que propôs um modelo baseado em redes neurais que classifica serpentes peçonhentas e não peçonhentas.

1.2 Objetivos

Este trabalho propõe um classificador que utiliza técnicas de redes neurais para identificar serpentes com base no interesse em saúde, a fim de agilizar o diagnóstico em casos de acidentes ofídicos, proporcionando às vítimas e aos profissionais da saúde uma ferramenta rápida e eficaz. Essa abordagem visa minimizar o intervalo entre o acidente e o início do tratamento adequado, potencialmente reduzindo complicações e melhorando os resultados de saúde. Além de servir de base para estudos futuros sobre a fauna brasileira. Para isso, visamos os seguintes objetivos específicos:

1. Coleta e disponibilização de um conjunto de dados específico;
2. Implementação e treino de modelos de aprendizado de máquina;
3. Análise dos resultados e identificação da melhor arquitetura;
4. Disponibilização dos resultados para a comunidade.

1.3 Organização

O texto está organizado em seis capítulos, os quais são brevemente descritos na sequência. No próximo capítulo, são apresentados conceitos fundamentais deste estudo, começando com aspectos biológicos e avançando para o funcionamento e avaliação de redes neurais, incluindo exemplos práticos. O Capítulo 2 faz uma revisão da literatura sobre o uso de redes neurais no reconhecimento de animais e discute conjuntos de dados disponíveis. No Capítulo 4, detalharemos a metodologia, abordando a seleção da arquitetura da rede, o pré-processamento de imagens e a construção do classificador. Os resultados são apresentados no Capítulo 5, com análises e discussões sobre o desempenho do modelo. Por fim, o Capítulo 6 reflete sobre as conclusões e limitações do estudo; e sugere direções para pesquisas futuras.

2 CONCEITOS BÁSICOS

Esse capítulo é dividido em duas seções: a primeira apresenta conceitos sobre serpentes de interesse em saúde pública no Brasil, incluindo definição e principais gêneros responsáveis por acidentes ofídicos. A segunda seção apresenta conceitos relacionados a redes neurais, em particular com suas definições, arquiteturas, métricas de avaliação, *transfer learning*, métodos de amostragem e hiperparâmetros.

2.1 Serpentes de interesse em saúde pública

Acidentes ofídicos, ou simplesmente ofidismo, é o quadro clínico decorrente da mordedura de serpentes (MINISTÉRIO DA SAÚDE, s.d.). Popularmente no Brasil, as serpentes são frequentemente chamadas de "cobras", mas tecnicamente esse termo é mais apropriado para descrever as serpentes da família Elapidae, como as corais verdadeiras. Algumas serpentes possuem glândulas venenosas capazes de produzir peçonha que interfere nos processos fisiológicos e bioquímicos, causando uma gama de efeitos adversos, incluindo reações colinérgicas, hemorrágicas, anticoagulantes, necróticas, miotóxicas, citolíticas e inflamatórias (MINISTÉRIO DA SAÚDE, s.d.). Dentre as espécies de serpentes peçonhentas, aquelas que são de interesse em saúde pública pertencem à duas famílias: Viperidae e Elapidae (MINISTÉRIO DA SAÚDE, s.d.).

Os acidentes causados por estas serpentes são categorizados em quatro tipos principais, baseados no gênero da serpente responsável (MINISTÉRIO DA SAÚDE, s.d.):

Acidente botrópico: Este tipo de acidente é provocado por serpentes dos gêneros *Bothrops* e *Bothrocophias*, pertencentes à família Viperidae. Estas serpentes, que incluem espécies como jararacuçu, jararaca, urutu, caiçaca e comboia, são encontradas em uma variedade de habitats em todo o Brasil. Representam a maioria dos acidentes ofídicos no país devido à sua ampla distribuição geográfica e diversidade de espécies.

Acidente crotálico: Causado pelas cascavéis, especificamente a espécie *Crotalus durissus* da família Viperidae. As cascavéis são facilmente identificáveis pela presença de um guizo ou chocalho na cauda. Habitam principalmente em cerrados, regiões áridas e semiáridas, campos e áreas abertas.

Acidente laquéutico: Este tipo de acidente é causado pela *Lachesis muta*, também da família Viperidae. Conhecida como surucucu-pico-de-jaca, é a maior serpente peçonhenta do Brasil e habita a floresta Amazônica e os remanescentes da Mata Atlântica.

Acidente elapídico: Causado por corais verdadeiras, pertencentes à família Elapidae e aos gêneros *Micrurus* e *Leptomicrurus*. Estas serpentes, que são encontradas em várias regiões do país, possuem um padrão característico de anéis coloridos.

Além desses, existem outros tipos de serpentes que podem causar acidentes ou envenenamentos, mas geralmente sem gravidade. Um exemplo são algumas serpentes da família Colubridae, que podem imitar a aparência das corais verdadeiras e são conhecidas como falsas corais. Apesar de possuírem glândulas de veneno, os envenenamentos causados por essas falsas corais geralmente não são considerados significativos do ponto de vista da saúde pública.

2.2 Redes Neurais Artificiais

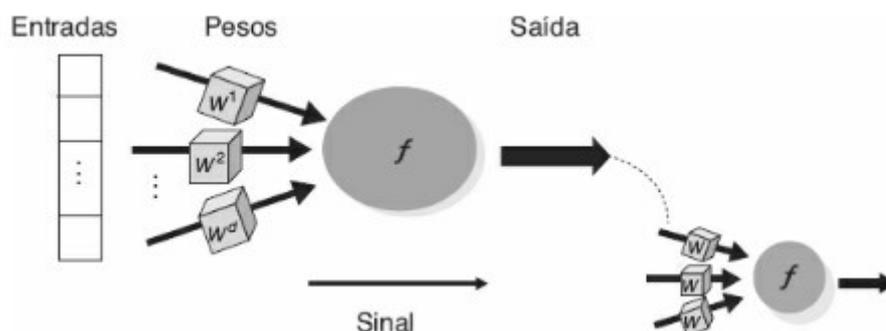
No dia a dia, executamos várias atividades que exigem foco em múltiplos eventos simultaneamente e processamento de informações diversas para agir corretamente. Atividades simples como segurar um objeto ou caminhar envolvem habilidades complexas como memória, aprendizado e coordenação motora. A dificuldade em programar robôs para realizar essas tarefas evidencia sua complexidade. Inspiradas pelo sistema nervoso e pelo cérebro humano, as Redes Neurais Artificiais (*Artificial Neural Network* - ANN) foram desenvolvidas para simular a capacidade humana de aprender e adquirir conhecimento (FACELI et al., 2021).

As ANNs são compostas por um conjunto densamente interconectado de unidades de processamento, chamadas de neurônios artificiais, sendo organizadas em uma ou mais camadas interligadas com o intuito de processar informações, aprender padrões e tomar decisões (FACELI et al., 2021).

Os neurônios de uma camada processam entradas aplicando pesos e usando uma função de ativação para determinar o sinal de saída, enviando, assim, essa saída para a próxima camada. O aprendizado é alcançado ajustando esses pesos com base nos dados de treinamento para minimizar o erro na previsão ou classificação final realizada pela rede (FACELI et al., 2021). Na Figura 2.1 podemos observar um esquema evidenciando a conexão entre dois neurônios.

Em uma ANN multicamadas, a primeira camada é a camada de entrada, sendo responsável por receber os dados iniciais para o processamento. As camadas intermediárias, conhecidas como camadas ocultas, são responsáveis pelo processamento complexo dos dados, realizando cálculos e transformações para extrair características e padrões. A

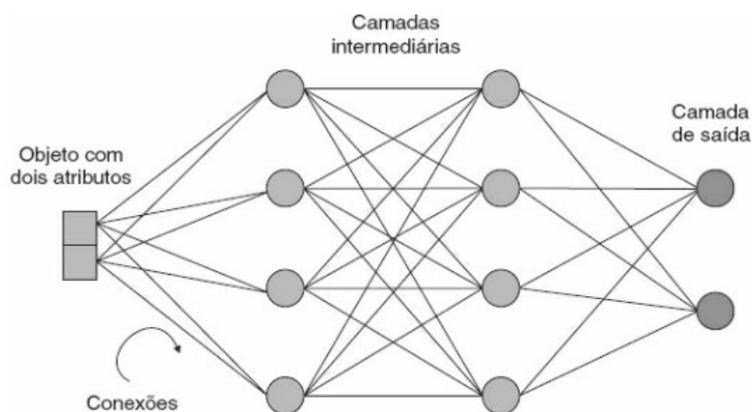
Figura 2.1 – Representação esquemática de um modelo simplificado de neurônio artificial.



Fonte: (FACELI et al., 2021).

última camada, a camada de saída, fornece o resultado final do processamento da rede, como uma classificação ou uma previsão. Tal comportamento está representado na Figura 2.2.

Figura 2.2 – Representação esquemática de uma Rede Neural Artificial multicamadas, mostrando a camada de entrada com dois atributos, múltiplas camadas intermediárias de neurônios e uma camada de saída, todas interligadas por conexões.



Fonte: (FACELI et al., 2021).

Existem dois tipos de problemas de classificação de acordo com o número de classes: a classificação binária, onde existem apenas duas classes; e classificação multi-classe, onde o número de classes é maior que dois. Neste trabalho, focamos na classificação de serpentes como sendo de interesse médico ou não, o que configura um cenário com apenas duas categorias distintas. Portanto, estamos lidando com um problema de classificação binária.

2.2.1 Rede Neural Convolutacional

A Rede Neural Convolutacional (*Convolutional Neural Network* - CNN) é um tipo de arquitetura de rede neural de sucesso em diversas áreas, incluindo mas não limitado a visão computacional e processamento de linguagem natural (LI et al., 2022). Além disso, esta área tem atraído a atenção tanto do meio industrial quanto do meio acadêmico (LI et al., 2022).

Uma CNN processa imagens através de camadas convolutivas que utilizam filtros (*kernels*) para detectar características como bordas e formatos. Estes filtros são aplicados em toda a imagem para produzir mapas de características (GOODFELLOW; BENGIO; COURVILLE, 2016).

As camadas de *pooling* seguem, reduzindo a dimensionalidade dos dados ao manter somente os atributos mais significativos. Esses processos ajudam a CNN a processar as imagens de maneira eficiente, preparando os dados para as camadas totalmente conectadas, que integram as informações para a classificação final ou reconhecimento de padrões (LI et al., 2022). Para o nosso contexto, diferentes serpentes podem ser identificadas através de características como formato da cabeça, formato do corpo, textura do corpo, cor das escamas e formato de olhos (ABDURRAZAQ; SUYANTO; UTAMA, 2019); traços estes que podem ser reconhecidos por diferentes camadas.

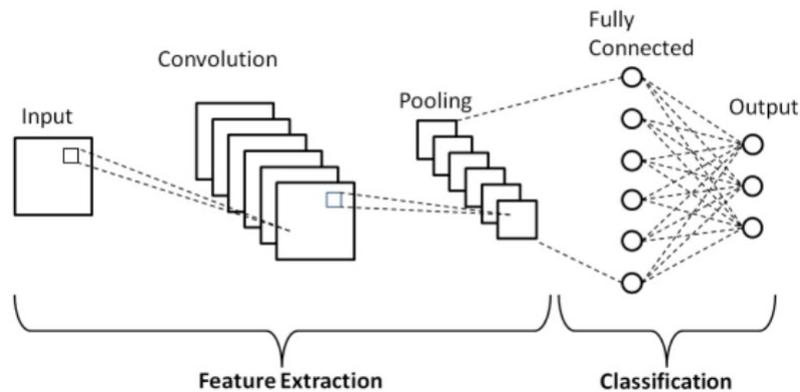
Por fim, as últimas camadas destas redes encontram-se camadas totalmente conectadas. Neste estágio, cada neurônio artificial de uma camada está interligado com todos os neurônios da camada subsequente. Este arranjo assegura que cada valor contribua para o processo de classificação da imagem, integrando todas as características identificadas nas etapas anteriores da rede (HUSSAIN; BIRD; FARIA, 2019). A Figura 2.3 ilustra as etapas de uma CNN.

2.2.2 Arquiteturas

Dentre as diferentes arquiteturas de ANNs propostas, neste trabalho lidaremos com as seguintes:

VGG: A VGG (*Visual Geometry Group*) (SIMONYAN; ZISSERMAN, 2015) caracteriza-se pelo uso de várias camadas convolucionais com filtros pequenos (3×3) e pelo aumento da profundidade da rede, com até 19 camadas. A VGG se destacou por sua simplicidade e eficácia, alcançando resultados notáveis em competições de visão com-

Figura 2.3 – Representação esquemática de uma CNN, mostrando a camada convolutiva para detectar características, as camadas de *pooling* e as camadas finais totalmente conectadas.

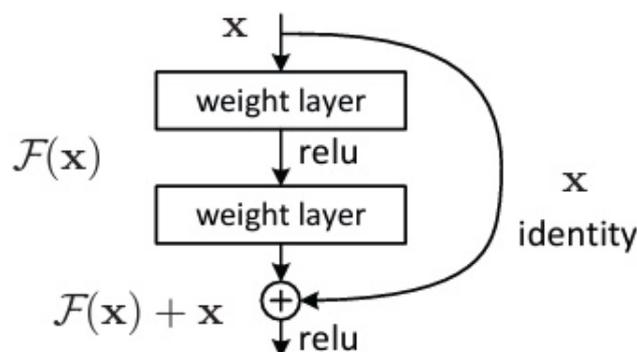


Fonte: (PHUNG; RHEE, 2019).

putacional como a *ImageNet Challenge 2014* (RUSSAKOVSKY et al., 2015). Ela usa camadas de *max-pooling* para redução de dimensionalidade e várias camadas totalmente conectadas no final.

ResNet: A arquitetura ResNet (HE et al., 2015) foi desenvolvida para resolver o problema de degradação do treinamento em redes neurais profundas. Esse problema ocorre quando conforme a rede se torna mais profunda, a eficiência do treinamento diminui e a rede começa a ter um desempenho pior, mesmo com o aumento do número de camadas. Isso ocorre devido a questões como o desaparecimento do gradiente, que dificulta a atualização efetiva dos pesos nas camadas mais iniciais da rede durante o processo de aprendizado (HE et al., 2015). A ResNet aborda isso introduzindo o aprendizado residual, uma técnica que agrupa camadas em blocos residuais com conexões de atalho (*shortcut connections*) e utiliza a soma da entrada com o valor processado pelo bloco como saída. Este comportamento está representado na Figura 2.4.

Figura 2.4 – Esquema de um bloco residual com duas camadas e a conexão de atalho.

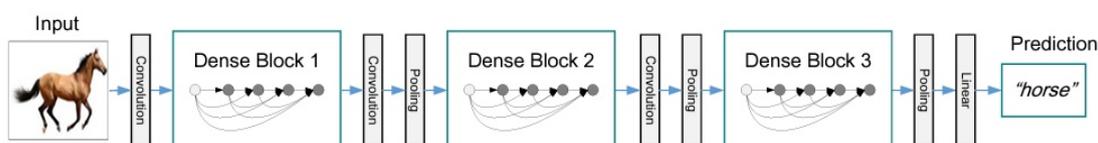


Fonte: (HE et al., 2015).

DenseNet: A DenseNet (HUANG et al., 2017) propõe uma solução para o problema de degradação ao criar blocos densos. Eles são compostos de conjuntos de camadas onde cada camada recebe entradas de todas as camadas anteriores dentro do mesmo bloco. Isso cria uma rede convolucional densamente conectada, onde as informações e características de cada camada são passadas para todas as outras camadas subsequentes.

Um exemplo de organização e comportamento da DenseNet pode ser visto na Figura 2.5.

Figura 2.5 – Esquema de uma DenseNet com três blocos densos.



Fonte: (HUANG et al., 2017).

Inception V3: A Inception V3 (SZEGEDY et al., 2016) aprimora a eficiência computacional em relação às suas versões anteriores. Isso é alcançado através da utilização de *factorized convolutions*, que substituem convoluções maiores (como filtros 5×5 ou 7×7) e mais custosas por sequências de camadas menores (como filtros 3×3) e mais eficientes. Além disso, a Inception V3 incorpora estratégias para evitar o *overfitting*, fenômeno que ocorre quando um modelo se ajusta excessivamente aos dados de treinamento, capturando detalhes e ruídos que não se generalizam bem para novos dados. Para isso, foi introduzido o *label-smoothing regularization*, uma técnica que ajuda a tornar o modelo menos confiante nas suas previsões.

As arquiteturas mais promissoras são VGG, ResNet e DenseNet e MobileNet, considerando um estudo feito por Ahmed et al. (AHMED; GAD; ABOUTABL, 2023). Neste estudo, quatro modelos de CNN, incluindo VGG16, ResNet50, MobileNetV2 e DenseNet121, foram testados em diferentes cenários com várias classes. Os resultados mostraram que VGG16, DenseNet121 e MobileNetV2 obtiveram os melhores resultados em termos de precisão, com VGG16 alcançando a maior precisão de 97,09% em uma configuração com 45 classes (AHMED; GAD; ABOUTABL, 2023) enquanto que a Resnet teve a menor acurácia com 53,79% classificando 5 classes. No trabalho de Othman et al. (OTHMAN et al., 2021) a Inception V3 é usada também para a classificação de duas espécies de serpentes da Malásia e atingiu uma acurácia de 90,0%.

2.2.3 Transfer learning

A transferência de aprendizado (*transfer learning*) é uma técnica que visa melhorar o desempenho de um modelo em uma tarefa-alvo, utilizando conhecimentos adquiridos em tarefas anteriores. Este conceito é inspirado pela capacidade humana de aplicar habilidades aprendidas em um contexto para resolver problemas em outro. Ao invés de treinar o modelo do zero para uma dada tarefa, os pesos aprendidos anteriormente são aproveitados para o novo modelo, economizando tempo e processamento (BORWANGINN et al., 2021).

Geralmente, são adicionadas novas camadas ao final da rede neural pré-treinada (YOSINSKI et al., 2014). Há a possibilidade de manter os pesos das primeiras camadas fixos, concentrando-se no treinamento das novas camadas adicionadas. Essa estratégia tem o propósito de aproveitar as características aprendidas pelas camadas pré-treinadas e aperfeiçoar as capacidades do modelo para uma tarefa específica.

Como alternativa, pode-se optar por deixar que aconteça o *backpropagation* por toda a rede, proporcionando o *fine-tuning* dela (YOSINSKI et al., 2014).

2.2.4 Métricas de avaliação

As métricas de avaliação de redes neurais oferecem meios para entender o desempenho de modelos preditivos. Essas métricas, como acurácia (*accuracy*), precisão (*precision*), sensibilidade (*recall*) e *F1-score*, fornecem uma visão detalhada sobre como um modelo lida com diferentes tipos de dados, resultando em perspectivas diferentes sobre o desempenho do modelo (THARWAT, 2020).

A matriz de confusão é uma ferramenta fundamental na avaliação de modelos de aprendizado de máquina, pois fornece uma visão do desempenho do modelo. Nela, um dos eixos representa o valor real dos dados e o outro indica o valor previsto pelo modelo. As métricas de avaliação são calculadas a partir dos valores presentes na matriz (Figura 2.6), valores tais como:

- Verdadeiros Positivos (VP) (*True Positive (TP)*): Representa o número de casos em que o modelo corretamente previu a classe positiva.
- Falsos Positivos (FP) (*False Positive (FP)*): Representa o número de casos em que o modelo previu a classe positiva, mas a verdadeira classificação é negativa.

Figura 2.6 – Figura ilustrativa de uma matriz de confusão para um problema de classificação binária.

		True/Actual Class	
		Positive (P)	Negative (N)
Predicted Class	True (T)	True Positive (TP)	False Positive (FP)
	False (F)	False Negative (FN)	True Negative (TN)
		P=TP+FN	N=FP+TN

Fonte: (THARWAT, 2020).

- Verdadeiros Negativos (VN) (*True Negative (TN)*): Representa o número de casos em que o modelo corretamente previu a classe negativa.
- Falsos Negativos (FN) (*False Negative (FN)*): Representa o número de casos em que o modelo previu incorretamente a classe negativa, mas a verdadeira classificação é positiva.

A partir da matriz de confusão, podemos extrair métricas de avaliação, tais como:

Accuracy: Mede a proporção total de previsões corretas feitas por um modelo de aprendizado de máquina, incluindo tanto as previsões verdadeiramente positivas quanto as verdadeiramente negativas, em relação ao total de previsões feitas, ou seja, ela mede a eficácia do modelo em classificar corretamente tanto as classes positivas quanto as negativas (THARWAT, 2020). O cálculo da métrica é dado através da Equação 2.1.

$$\text{Accuracy} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.1)$$

Precision: Mede a proporção de previsões verdadeiramente positivas em relação ao total de previsões positivas feitas pelo modelo. Em outras palavras, é a fração dos casos identificados como positivos pelo modelo que são de fato positivos. A precisão é especialmente importante em situações onde os custos dos falsos positivos são altos (THARWAT, 2020). Seu cálculo é dado através da Equação 2.2.

$$\text{Precision} = \frac{VP}{VP + FP} \quad (2.2)$$

Recall (Sensibilidade): Mede a proporção de previsões verdadeiramente positivas

em relação ao total de casos positivos reais. É a capacidade do modelo de identificar corretamente todas as instâncias positivas. O *recall* é muito útil em situações onde é de grande importância detectar todos os casos positivos, como em diagnósticos médicos, onde perder um caso positivo pode ter consequências graves (THARWAT, 2020). O cálculo desta métrica é realizado utilizando a seguinte Equação 2.3.

$$\text{Recall} = \frac{VP}{VP + FN} \quad (2.3)$$

F1-score: Métrica que combina precisão e *recall* em um único número, oferecendo um balanço entre essas duas métricas. É o harmônico médio de precisão e *recall*, proporcionando uma medida de qualidade geral do modelo, especialmente útil quando se deseja um equilíbrio entre a capacidade de identificar corretamente as instâncias positivas (*recall*) e a precisão dessas identificações (THARWAT, 2020). Seu cálculo é dado pela Equação 2.4.

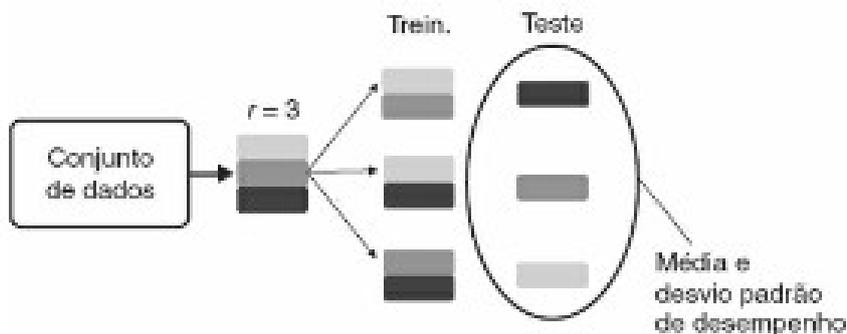
$$\text{F1-score} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.4)$$

2.2.5 Métodos de avaliação de modelos

Para garantir métricas de avaliação confiáveis, é essencial dividir os dados em subconjuntos distintos para treinamento e teste. Essa separação ajuda a assegurar que o modelo seja eficaz não apenas nos dados usados no treinamento, mas também em novos dados. Existem diversas técnicas para realizar essa divisão de dados. Dentre elas, podemos citar:

K-fold cross-validation: Uma técnica usada para comparação de modelos preditivos, onde o conjunto de dados é dividido em K subconjuntos (ou *folds*). O modelo é treinado K vezes, cada vez usando um subconjunto diferente como conjunto de teste e os restantes como conjunto de treinamento, gerando um modelo para cada subconjunto diferente. Após cada rodada de treinamento e teste, avalia-se o desempenho do modelo. Por fim, a performance do modelo é calculada pela média dos desempenhos obtidos em cada uma das K iterações. Esta técnica é particularmente útil para obter uma avaliação mais confiável e menos enviesada do modelo, especialmente em conjuntos de dados menores ou desequilibrados (FACELI et al., 2021). Essa técnica é ilustrada na Figura 2.7.

Figura 2.7 – Figura ilustrativa de um *3-fold cross-validation*, evidenciando os subconjuntos de treino e de teste.



Fonte: (FACELI et al., 2021).

2.2.6 Hiperparâmetros

Hiperparâmetros são configurações essenciais na construção e otimização de modelos de aprendizado de máquina. Eles diferem dos parâmetros do modelo que são aprendidos durante o treinamento, pois devem ser definidos antecipadamente. Hiperparâmetros incluem escolhas como otimizador, taxa de aprendizado, número de camadas em uma rede neural, número de épocas de treinamento e tamanho do lote (*batch size*) usado para treinamento. A otimização de hiperparâmetros é um passo crucial, pois influencia diretamente o desempenho do modelo. Diferentes técnicas de otimização podem ser aplicadas para encontrar a configuração ideal de hiperparâmetros, melhorando assim a eficácia do modelo em tarefas específicas (YANG; SHAMI, 2020).

Otimizador: Ele é responsável pela determinação de como os pesos do modelo são atualizados durante o treinamento. A escolha do otimizador pode afetar a velocidade e a qualidade da convergência do modelo para um ótimo local ou global. Diferentes otimizadores, como *Stochastic Gradient Descent* (SGD) (SUTSKEVER et al., 2013), *Adaptive Moment Estimation* (Adam) (KINGMA; BA, 2014) ou *Root Mean Square Propagation* RMSprop (TIELEMAN; HINTON et al., 2012), têm características únicas que os tornam mais adequados para tipos específicos de problemas de aprendizado de máquina. A escolha correta do otimizador é essencial para garantir o treinamento eficiente e eficaz do modelo. O otimizador Adam combina elementos do AdaGrad (DUCHI; HAZAN; SINGER, 2011) e do RMSProp, com o objetivo de adaptar a taxa de aprendizado para cada parâmetro individual do modelo, se fazendo valer do conceito de *momentum* (SUTSKEVER et al., 2013), cuja técnica utiliza os últimos valores de atualização do gradiente para acelerar o otimizador nas direções consistente e reduzir oscilações.

Épocas de treinamento: Refere-se ao número de passagens completas por todo o conjunto de dados de treinamento. A duração do treinamento em épocas depende do tamanho do conjunto de dados e deve ser ajustada aumentando-se gradualmente seu valor até que a precisão de validação comece a diminuir, indicando *overfitting*. Frequentemente, modelos convergem após algumas épocas; e épocas adicionais podem levar a tempo de execução desnecessário e *overfitting*. Isso pode ser evitado usando o método de parada antecipada, que interrompe o treinamento quando a precisão ou outra métrica de validação de interesse não aumenta após um certo número de épocas consecutivas.

Taxa de aprendizagem: Ela determina o tamanho do passo a cada iteração, o que permite a convergência da função objetivo. Uma taxa de aprendizado alta acelera o processo de aprendizagem, mas pode causar oscilações em torno de um mínimo local ou até mesmo impedir a convergência. Por outro lado, uma taxa de aprendizado baixa converge suavemente, mas aumenta significativamente o tempo de treinamento do modelo, exigindo mais épocas de treinamento. Uma taxa de aprendizado adequada deve permitir que a função objetivo convirja para um mínimo global em um tempo razoável (YANG; SHAMI, 2020).

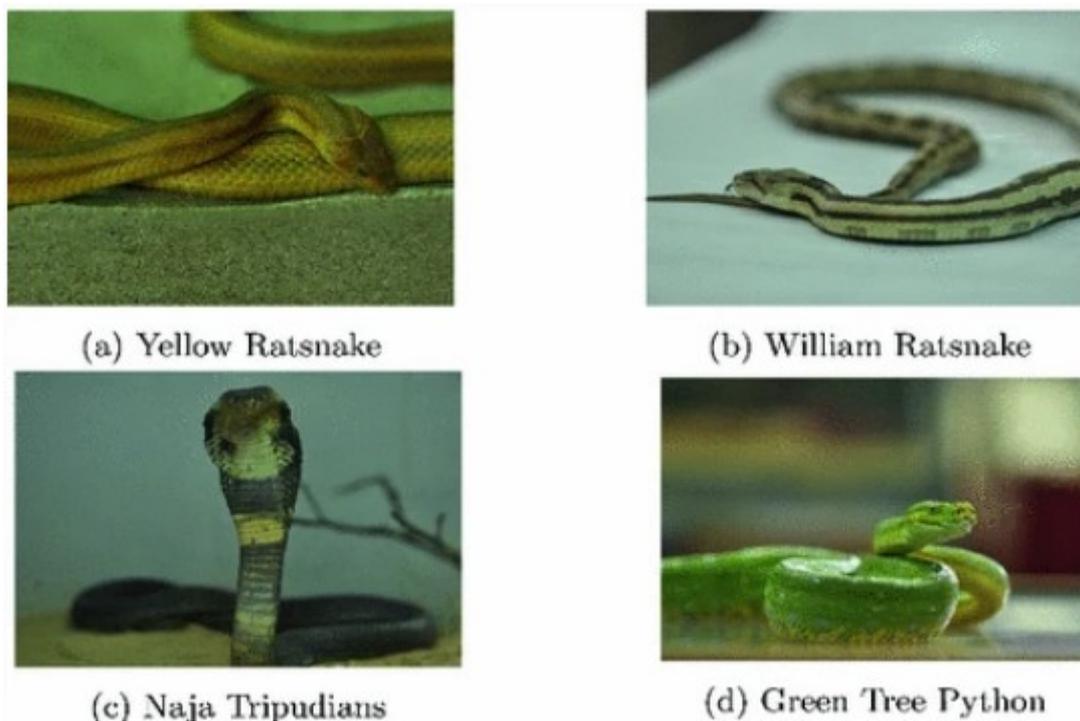
3 TRABALHOS RELACIONADOS

A classificação de imagens usando redes neurais é uma área que combina técnicas de Visão Computacional. Neste campo, algoritmos são treinados para identificar e categorizar objetos ou características em imagens digitais. Alguns exemplos incluem a identificação de rostos (HANGARAGI; SINGH; N, 2023), o reconhecimento de sinais de trânsito (LEE; KIM, 2019), identificação de sinais de doenças de pele (ALLUGUNTI, 2022), detecção de objetos proibidos em sistemas de segurança (GAUS et al., 2019), entre outros.

A identificação de espécies de serpentes é um desafio devido a vários fatores. O número de espécies conhecidas está crescendo rapidamente, com mais de 3800 espécies registradas e novas descobertas frequentes, impulsionadas pela globalização e avanços tecnológicos (DURSO et al., 2021). Mimetismo de espécies peçonhentas, na qual uma serpente de espécie não peçonhenta desenvolve a aparência de uma espécie peçonhenta. O mimetismo das corais verdadeiras (peçonhentas) (RABOSKY et al., 2016). Todos estes desafios leva à identificação de características de serpentes a ser de interesse médico e também interesse no âmbito da computação.

Em seguida apresentaremos os trabalhos mais relevantes sobre identificação de serpentes. O trabalho de Amir et al. (AMIR et al., 2017) propôs um modelo para classificação de espécies de serpentes, ele investigou a acurácia de cinco técnicas de aprendizado de máquina: *nearest neighbors*, *k-nearest neighbors (KNN)*, *Decision tree J48*, *naive Bayes* e *backpropagation neural network*. O *dataset* de imagens chamado de *Snakes of Perlis Corpus* (Figura 3.1) contém 349 imagens de 22 espécies que podem ser encontradas na Malásia. A espécie com maior número de amostras é referenciada como "William Ratsnake" e possui 33 imagens, enquanto que a espécie com o menor número de imagens é a *Bungarus candidus* com 3 imagens. Os resultados mostraram que o *nearest neighbors* e a *backpropagation neural network* obtiveram as maiores acurácias com 89,22% e 87,93% respectivamente; enquanto que o KNN com K=7 obteve 80,34%, Naive Bayes com 75,64% e *Decision tree J48* com a menor acurácia, 71,29%.

O trabalho de Ahmed et al. (AHMED; GAD; ABOUTABL, 2023) utilizou quatro modelos CNN (VGG16, ResNet50, MobileNetV2 e DenseNet121) para identificar espécies de serpentes. Foram conduzidos experimentos usando 5, 10, 16, 20, 22 e 45 espécies e épocas de treinamento variando entre 20 e 35. Um dos diferenciais é a utilização do *Salient Object Detection (SOD)* para detectar a presença de serpentes em imagens com

Figura 3.1 – Exemplo de imagens no *dataset Snakes of Perlis Corpus*

Fonte: (AMIR et al., 2017).

um fundo complexo, ou seja, quando elas estão camufladas. A partir disso, os modelos de classificação anteriormente são aplicados. O modelo com o melhor desempenho foi o VGG16, com acurácia de 99,32%, 97,12%, 98,53%, 97,93%, 98,01%, 97,09% para 5, 10, 16, 20, 22 e 45 espécies respectivamente.

Abyaratne, Ilmini and Fernando (2019) propuseram um modelo para classificação de espécies de serpentes no qual cinco modelos CNN foram treinados para classificar seis espécies de serpentes no Sri Lanka. O *dataset* contém 250 imagens para cada classe que passaram pelo processo de *data augmentation*, processo utilizado para aumentar o volume, qualidade e diversidade dos dados de treinamento (MUMUNI; MUMUNI, 2022), processo este que incrementou o número de imagens a 2000 por classe, que foram divididas 60% para treino, 20% para validação e 20% para teste. Quatro dos modelos CNN foram aplicados utilizando arquiteturas pré-treinadas: ResNet50, VGG16, MobileNet e InceptionV3, enquanto que um modelo foi desenvolvido do zero. O resultado indicou que a arquitetura MobileNet obteve a melhor acurácia com 90,5%.

O trabalho de Progga et al. (2021) propôs um modelo baseado em CNN que classifica serpentes de duas classes: peçonhentas e não-peçonhentas. O *dataset* utilizado possuía 1766 imagens de serpentes. A técnica de *transfer learning* foi utilizada com sete modelos de ANNs para aumentar a acurácia do modelo, que por sua vez atingiu uma

acurácia de 91,30% usando *cross validation* com cinco *folds*.

Kalinathan et al. (2021) propuseram um sistema para classificação automática de serpentes. Este sistema auxilia na classificação de 772 espécies de serpentes usando o conjunto de dados SnakeCLEF 2021, que contém 412.537 imagens. O modelo ResNeXt50-V2 foi utilizado no processo de classificação e alcançou uma precisão de 85,7%.

Vasmatkar et al. (2020) propuseram um sistema baseado em CNN para classificar espécies de serpentes. O conjunto de dados contém 3050 imagens de 28 espécies de serpentes, e três arquiteturas de CNN, incluindo VGG16, MobileNet e DenseNet, foram usadas no processo de classificação. Os resultados indicaram que a precisão do VGG16, MobileNet e DenseNet foi de 12,28%, 58,65% e 72%, respectivamente.

Dube and Bhuru (2022) desenvolveram um aplicativo móvel para classificar espécies de serpentes com o auxílio da potência de CNN. O conjunto de dados usado neste estudo foi coletado da internet a partir de instituições como a *African Snakebite Institute* (Figura 3.2) e inclui oito espécies de serpentes. As imagens no conjunto de dados foram anotadas usando uma ferramenta online de anotação de dados chamada *MakeSense.ai*. O modelo YOLOv5 foi utilizado para o treinamento e alcançou uma precisão de 71%.

Figura 3.2 – Exemplo de imagens no *dataset* usado por Dube et al. (DUBE; BHURU, 2022).

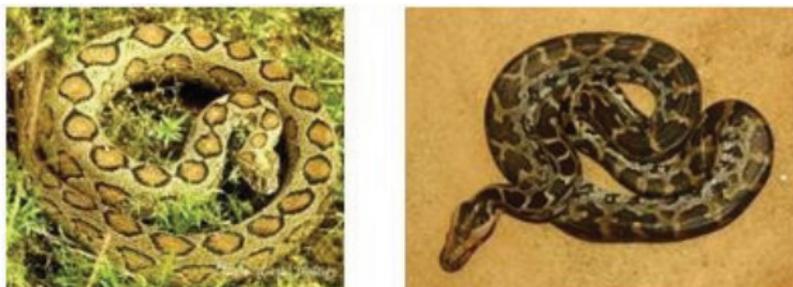


Fonte: (DUBE; BHURU, 2022).

Dandeniya, Wickramasinghe and Dasanayaka (2022) apresentaram uma aplicação web baseada em CNN e transformer para classificar espécies de serpentes. Um método de conjunto é usado para cinco modelos de CNN, incluindo: EfficientNet B0, EfficientNet B7, ResNet50, ResNet100, Data-Efficient Image Transformers. O modelo foi avaliado em um conjunto de dados que contém apenas duas classes: Russell's Viper e Indian Rock

Python (Figura 3.3); ele contém 500 imagens para cada classe, então o conjunto de dados inteiro contém apenas 1000 imagens. Ele é dividido em 60%, 20% e 20% para treinamento, validação e teste, respectivamente. A precisão de teste obtida foi de 94,5%. Apesar disso, o número de imagens e classes é muito baixo.

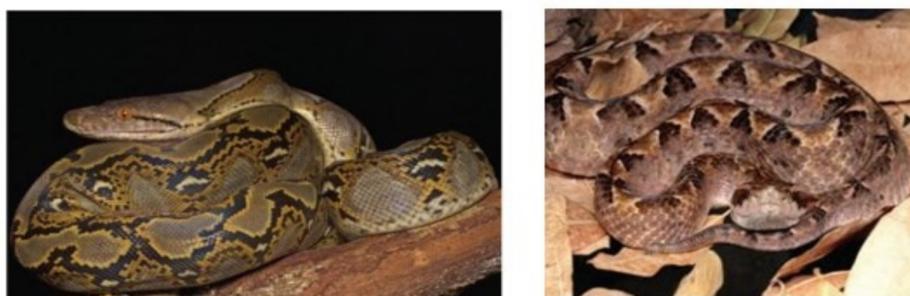
Figura 3.3 – Exemplo de serpentes *Russell's Viper* e *Indian Rock Python*, respectivamente, utilizadas no trabalho de Dandeniya et al. (DANDENIYA; WICKRAMASINGHE; DASANAYAKA, 2022).



Fonte: (DANDENIYA; WICKRAMASINGHE; DASANAYAKA, 2022).

O trabalho de Othman et al. (2021) apresentou um modelo baseado na Inception V3 para classificação de duas espécies da Malásia: a *Malayopython reticulatus* e a *Calloselasma rhodostoma* (Figura 3.4). O *dataset* era composto de 240 imagens sendo 200 usadas para treino e 40 para teste e acurácia do modelo atingiu a marca dos 90,0%

Figura 3.4 – Exemplo de serpentes *Malayopython reticulatus* e *Calloselasma rhodostoma*, respectivamente, utilizadas no trabalho de Othman et al. (OTHMAN et al., 2021).



Fonte: (OTHMAN et al., 2021).

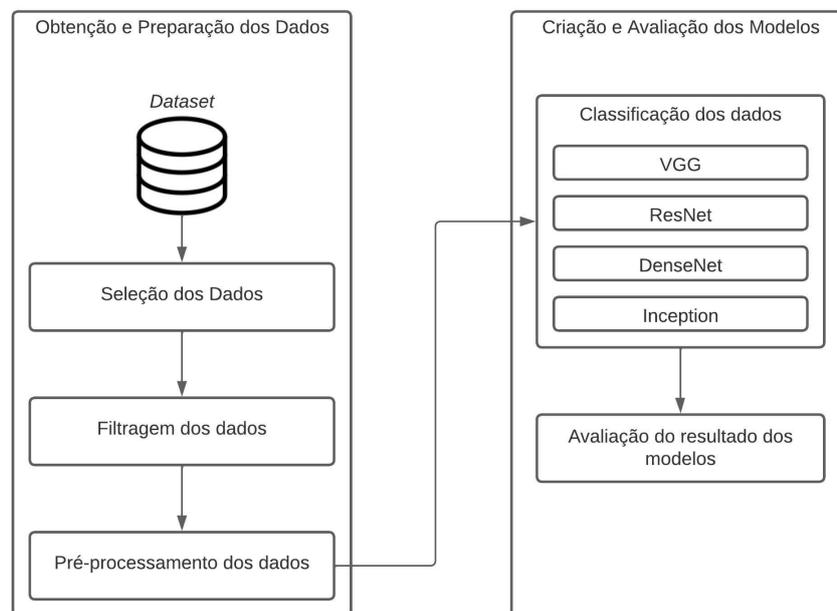
A partir da literatura, não encontramos nenhum trabalho de classificação com o intuito de identificar serpentes de interesse em saúde. Os trabalhos disponíveis apresentavam modelos para classificar espécies de serpentes. Com exceção do trabalho de Progga et al. (2021) que apresentou um modelo para classificar serpentes entre peçonhenta e não-peçonhenta. O volume dos *datasets* varia bastante entre os trabalhos apresentados, aqueles que possuem poucos dados utilizaram técnicas de *data augmentation* para lidar com essa dificuldade. Vale constatar que os trabalhos avaliados neste capítulo apresenta-

ram seus resultados com base na acurácia e precisão como métricas de desempenho. Uma vez que a acurácia é uma métrica sensível a conjunto de dados desbalanceados, usaremos também o *F1-score*.

4 METODOLOGIA

A Figura 4.1 ilustra as etapas seguidas por este trabalho. Ela passa pela criação do *dataset*, pré-processamento dos dados, criação de modelos de quatro arquiteturas CNN diferentes para classificar serpentes de interesse em saúde pública, validação sobre um conjunto de teste e avaliação dos diferentes modelos.

Figura 4.1 – Fluxograma da metodologia do trabalho desde a obtenção e criação do conjunto de dados à criação e avaliação dos modelos.



Fonte: O Autor.

4.1 Dataset

Para a realização deste trabalho, criamos um *dataset* através dos passos descritos a seguir:

1. Adquirir a Lista de Serpentes do Brasil: Extraída da lista de répteis do Brasil (GUEDES; ENTIAUSPE-NETO; COSTA, 2023) da ordem *Squamata*, subordem *Serpentes* (OSSIBOFF, 2018), usando como filtro todas as espécies de serpentes com ocorrência no Brasil de acordo com Guedes et al. (GUEDES; ENTIAUSPE-NETO; COSTA, 2023).
2. Adquirir imagens a partir de uma fonte de imagens de Serpentes: A GBIF (*Global*

Biodiversity Information Facility), uma rede internacional e infraestrutura de dados financiada pelos governos do mundo e tem como objetivo fornecer a qualquer pessoa, em qualquer lugar, acesso aberto a dados sobre todos os tipos de vida no planeta (GBIF: The Global Biodiversity Information Facility, 2024).

3. Encontrar a definição de serpentes de interesse em saúde pública do Brasil: Extraída a partir da definição dos acidentes ofídicos pelo Ministério da Saúde do Brasil (MINISTÉRIO DA SAÚDE, s.d.).
4. Construir o *Dataset* de serpentes de interesse em saúde pública do Brasil: Através de um *script* escrito na linguagem *Python*, foi criado um conjunto de imagens com todas as espécies presentes na fauna brasileira e presente nos dados da GBIF¹. Uma vez com as imagens coletadas, os dados foram separados entre "medico" e "nao-medico", indicando imagens com serpentes de interesse médico e interesse não-médico.

4.1.1 Filtragem de Dados

Originalmente, o *dataset* continha pastas de espécies de serpentes da fauna brasileira vazias, ou seja, o algoritmo não encontrou ocorrências de imagens no banco de imagens da GBIF. Além disso, haviam imagens que não puderam ser acessadas uma vez no *dataset*. A filtragem de dados foi feita com o objetivo de eliminar pastas de espécies vazias e imagens que não puderam ser abertas.

Após a filtragem de dados, o *dataset* contém 1863 imagens de serpentes de interesse em saúde pública e 7373 imagens de serpentes que não são de interesse em saúde pública, apesar de ser um conjunto de dados abundante, as classes estão desbalanceadas, com 20,2% do *dataset* sendo imagens de serpentes de interesse em saúde pública e 79,8% não sendo de interesse em saúde pública. Exemplo dessas imagens podem ser encontradas na Figura 4.2.

4.2 Pré-processamento das imagens

Foi necessário preparar as imagens de entrada antes de alimentá-las nas redes neurais. Primeiramente, todas as imagens foram redimensionadas para a resolução de 224×224

¹<https://github.com/daivesk/TCC_Serpentes/tree/main/imageSet>

Figura 4.2 – Exemplos de imagens presentes no *dataset*.(a) *Bothrops atrox*(b) *Lachesis muta*(c) *Boa constrictor*(d) *Lygophis anomalus*

Fonte: O Autor.

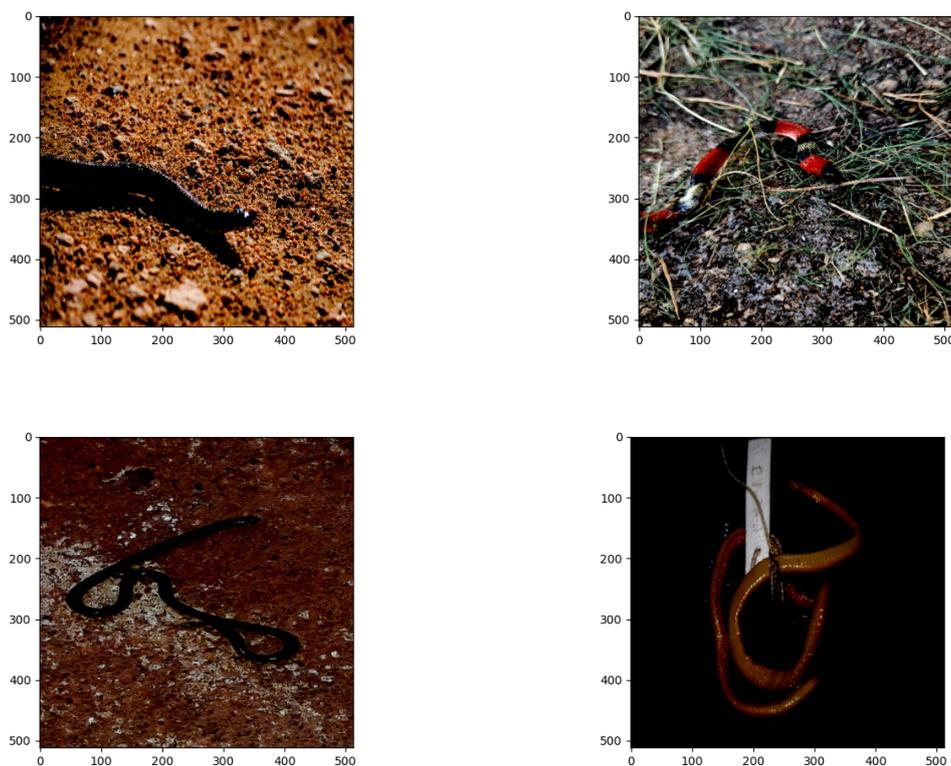
pixels, com o intuito de padronizar as entradas. Isto foi necessário pois as imagens do nosso conjunto de dados possuíam resoluções variadas. Este processo também acelera o processo de treinamento, pois a rede passa a ter menos parâmetros para aprender (SAPO-NARA; ELHANASHI, 2021).

A técnica de *zero-centering* foi aplicada em cada canal de cor das imagens. *Zero-centering* é uma técnica que ajusta o centro da distribuição dos pesos para zero antes da quantização minimizando a degradação de desempenho, enquanto mantém a precisão na detecção de objetos em redes neurais e aumenta a eficiência em recursos de *hardware*. A Figura 4.3 exemplifica algumas destas imagens.

Além destas alterações, algumas técnicas foram aplicadas por serem requisitos para o uso adequado de algumas arquiteturas escolhidas para a geração dos modelos, entre elas:

Alteração do modelo de cores RGB para BGR: A inversão da ordem dos canais de cores foi necessária para que modelos como VGG e ResNet possam usar configurações

Figura 4.3 – Exemplos de imagens pré-processadas.



Fonte: O Autor.

similares com as quais foram treinadas previamente.

Normalização dos canais de cores: A normalização dos canais de cores refere-se ao processo de ajustar os valores dos canais de cores em uma imagem para um determinado intervalo ou escala. No caso deste trabalho, isso foi feito para que os valores dos canais de cores ficassem entre zero e um. Essa técnica visa o aprendizado mais eficiente da DenseNet.

4.3 Classificadores

Durante o treinamento de modelos, quatro arquiteturas foram utilizadas: a VGG16 (SIMONYAN; ZISSERMAN, 2015), a ResNet (HE et al., 2016), a DenseNet (HUANG et al., 2017) e a Inception V3 (SZEGEDY et al., 2016).

Todos os experimentos foram feitos através da plataforma Google Colaboratory num ambiente de execução T4², que disponibiliza uma GPU (*Graphics Processing Unit*)

²<https://colab.research.google.com/github/d2l-ai/d2l-tvm-colab/blob/master/chapter_gpu_schedules/arch.ipynb>

Tesla T4³. A arquitetura tem como propósito a aceleração da criação de modelos de aprendizado de máquina.

Com o intuito de realizar uma comparação entre as arquiteturas, alguns hiperparâmetros foram mantidos, entre eles o otimizador ADAM com *learning rate* de 0,00001, enquanto que o *batch size* e o número de épocas de treinamento foram variados.

³<<https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf>>

5 RESULTADOS E DISCUSSÃO

Para a análise dos resultados, optamos por utilizar principalmente o *F1-score*. Acurácia também será referida com o objetivo de comparar com os resultados obtidos nos trabalhos relacionados. A decisão de usar o *F1-score* se deve ao desbalanceamento de dados. No caso do presente trabalho, a classe majoritária é a de serpentes que não são de interesse médico; portanto a acurácia pode representar uma performance que não condiz com o cenário prático. O *F1-score* também enfatiza Falsos Positivos e Falsos Negativos, para o intuito deste trabalho, Falsos Negativos são muito valiosos uma vez que representaria a predição equivocada de uma serpente que na realidade pode causar danos graves.

Os testes com *batch sizes* menores não apresentaram um bom desempenho, conforme pode ser visto nas Tabelas 5.1, 5.2, 5.3 e 5.4. Esse comportamento é esperado, uma vez que, além do *dataset* ser desbalanceado, a classe de interesse também tem o menor número de dados. Assim, o modelo pode não ter treinado adequadamente a classe minoritária. Vale ressaltar que a acurácia dos modelos tem uma diferença muito significativa em relação ao *F1-score*. A acurácia representa um desempenho regular enquanto que o *F1-score* salienta como os modelos lidam com Verdadeiros Positivos em relação aos Falsos Positivos e Falsos Negativos.

Tabela 5.1 – Desempenho das arquiteturas com *batch size* de 256 e 20 épocas de treinamento.

Modelo	Acurácia	F1-score
VGG	0,773	0,383
ResNet	0,773	0,340
DenseNet	0,750	0,272
Inception	0,804	0,341

Fonte: O Autor.

Tabela 5.2 – Desempenho das arquiteturas com *batch size* de 256 e 35 épocas de treinamento.

Modelo	Acurácia	F1-score
VGG	0,789	0,412
ResNet	0,777	0,296
DenseNet	0,793	0,131
Inception	0,750	0,304

Fonte: O Autor.

Quando aumentamos o *batch size*, os resultados demonstram uma melhora geral no desempenho, como pode ser visto nas tabelas 5.5 e 5.6.

Tabela 5.3 – Desempenho das arquiteturas com *batch size* de 512 e 20 épocas de treinamento.

Modelo	Acurácia	F1-score
VGG	0,820	0,438
ResNet	0,796	0,380
DenseNet	0,779	0,306
Inception	0,779	0,395

Fonte: O Autor.

Tabela 5.4 – Desempenho das arquiteturas com *batch size* de 512 e 35 épocas de treinamento.

Modelo	Acurácia	F1-score
VGG	0,798	0,455
ResNet	0,810	0,247
DenseNet	0,832	0,481
Inception	0,808	0,394

Fonte: O Autor.

A VGG com *batch size* 7500 e 35 épocas de treinamento obteve o melhor desempenho entre os experimentos feitos, evoluindo de um *F1-score* de 0,412 para 0,615 quando variamos o *batch size* e mantendo o número de épocas em 35, isso por ter acontecido pelo maior número de imagens de espécies de interesse em saúde disponível. O aumento de épocas de treinamento (mantendo o *batch size* fixo) não trouxe variações significantes, até piorando as métricas em algumas arquiteturas, isso pode ser devido ao *overfitting*. A acurácia da VGG com melhor desempenho se compara com o trabalho de Progga et al. (PROGGA et al., 2021) que propôs um modelo para classificar serpentes peçonhentas e não-peçonhentas, obtendo uma acurácia de 91,30% também usando *cross validation* com cinco *folds*.

Tabela 5.5 – Desempenho das arquiteturas com *batch size* de 7500 e 20 épocas de treinamento.

Modelo	Acurácia	F1-score
VGG	0,859	0,615
ResNet	0,857	0,581
DenseNet	0,855	0,481
Inception	0,845	0,551

Fonte: O Autor.

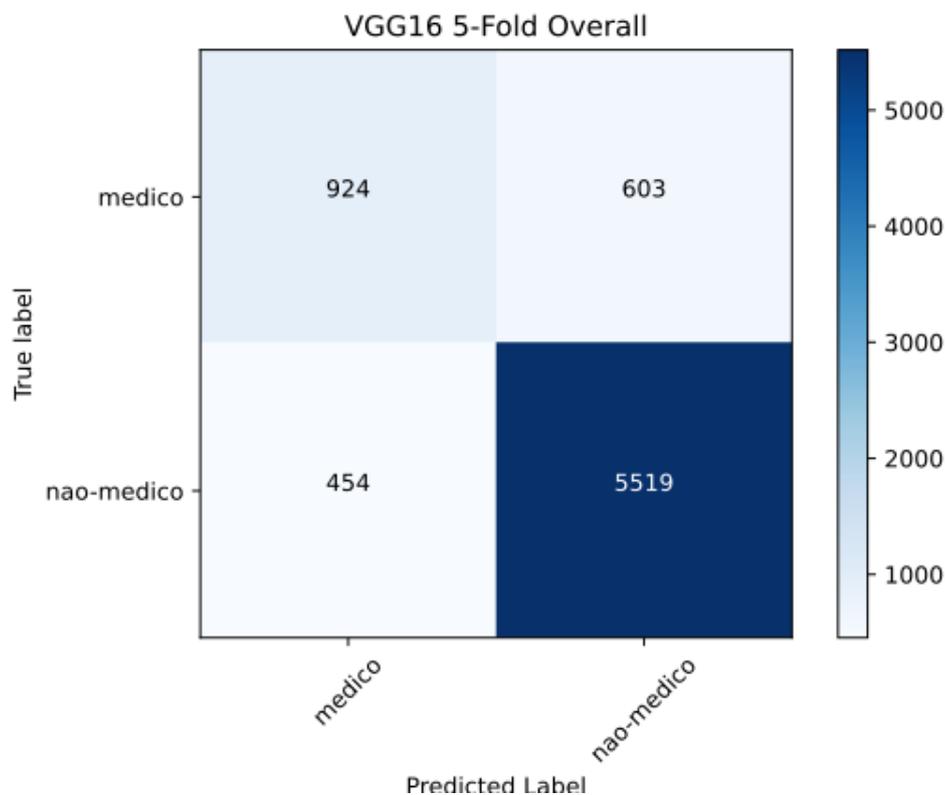
Apesar da acurácia, o classificador ainda tem um grande número de Falsos Negativos (Figura 5.1), podendo causar incidentes gravíssimos se fossem dados de cenários reais.

O valor do *F1-score* pode ser explicado, além do desbalanceamento dos dados, pelo mimetismo entre espécies, por exemplo, no *dataset* há a presença da espécie de interesse em saúde pública popularmente chamada de "Coral Verdadeira" (*Micrurus co-*

Tabela 5.6 – Desempenho das arquiteturas com *batch size* de 7500 e 35 épocas de treinamento.

Modelo	Acurácia	F1-score
VGG	0,859	0,635
ResNet	0,861	0,607
DenseNet	0,855	0,590
Inception	0,846	0,576

Fonte: O Autor.

Figura 5.1 – Matriz de confusão da validação cruzada na VGG com *batch size* de 7500 e 35 épocas de treinamento.

Fonte: O Autor.

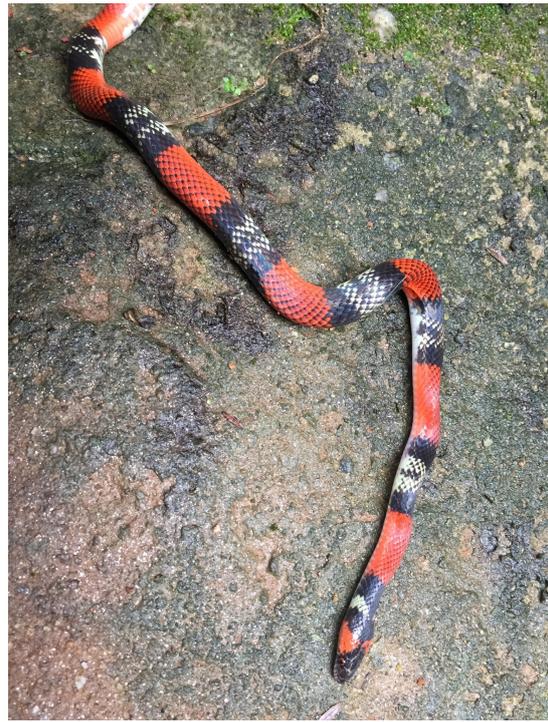
rallinus), mas também encontramos diversas variedades de espécies conhecidas como "Corais Falsas" (*Oxyrhopus guibei*, *Oxyrhopus trigeminus*, entre outras), estas não são de interesse em saúde pública, mas por serem muito similares (Figura 5.2), podem confundir o classificador.

É importante salientar que, por limitações do ambiente de execução utilizado, não foi possível treinar os modelos usando um *batch size* do tamanho total do *dataset*, ou seja, 9236 imagens. Apesar de não ter sido obtida uma performance de um classificador estado da arte, consideramos os resultados obtidos pelo classificador VGG com *batch size* de 7500 e 35 épocas de treinamento, satisfatórios, uma vez que, pelo que foi visto, este é o primeiro trabalho a propor uma abordagem para identificação de serpentes de interesse

Figura 5.2 – Exemplos de corais presentes na fauna brasileira.



(a) *Micrurus corallinus*



(b) *Oxyrhopus guibei*



(c) *Oxyrhopus trigeminus*

Fonte: O Autor.

em saúde pública, que abre caminho para futuras investigações.

6 CONCLUSÃO

Acidentes ofídicos podem ser prevenidos e ter seu tratamento agilizado com o auxílio da identificação de serpentes cujos acidentes envolvendo mordeduras podem evoluir para quadros graves. Este trabalho se dedicou a apresentar classificadores utilizando aprendizado de máquina para identificar serpentes de interesse em saúde pública com base em suas imagens. Para isso, criamos e disponibilizamos um conjunto de dados de serpentes da fauna brasileira disponibilizada pela Sociedade Brasileira de Herpetologia e da *Global Biodiversity Information Facility* (GBIF); e as serpentes de interesse em saúde pública foram obtidas através do Ministério da Saude do Brasil. O conjunto de dados foi utilizado para testar o desempenho de quatro arquiteturas previamente propostas para apresentar uma análise do melhor modelo para o classificador. O modelo com o melhor desempenho foi baseado na arquitetura VGG com um *F1-score* de 0,635 e acurácia de 0,859. Dessa forma, consideramos os resultados positivos, considerando o grande desbalanceamento no nosso *dataset*.

6.1 Limitações

Os resultados podem melhorar uma vez que mais dados forem adicionados ao *dataset*, isso inclui exemplares de imagens de serpentes de interesse em saúde pública, mas também exemplares que não são de interesse que estão presentes na fauna brasileira. Apesar possuir a lista de serpentes em território brasileiro, a GBIF não possuía o número mínimo de espécies desejado inicialmente, muitas das espécies sequer possuíam imagens presentes no banco de imagens consultado. O aumento de imagens pode trazer mais confiança aos resultados assim como prover mais características para que as arquiteturas possam estudar.

O desbalanceamento do *dataset* devido ao grande número de exemplares de não-interesse médico pode fazer com que o modelo tenda a prever o conjunto mais sobressalente. Trazendo um viés indesejado ao modelo final.

Apesar do *cross-validation* ser um bom método para avaliar a performance do modelo, a falta de um conjunto *holdout* pode fazer com o que as métricas sejam otimistas em relação a sua aplicação efetiva no mundo afora, ou seja, *overfitting*.

6.2 Trabalhos futuros

Com o objetivo de evoluir este trabalho, algumas melhorias poderiam ser realizadas. A principal adição seria a inclusão de dados sobre espécies cujas ocorrências não foram encontradas para este trabalho para uso tanto no treino quanto no teste da rede. Além disso, para mitigar o problema do desbalanceamento dos dados, processos de *data augmentation*, para aumentar o volume de dados de forma artificial poderia ser aplicado, por exemplo rotação, translação, espelhamento e *zoom* das imagens existentes.

Para um classificador que aproveite de todas as arquiteturas apresentadas, um *ensemble* poderia ser desenvolvido, neste modelo o resultado final é obtido pela combinação da decisão de diferentes preditores (FACELI et al., 2021). O resultado do preditor pode ser dado de formas diversas, entre elas a votação, na qual os preditores votam o conforme sua predição; os votos podem ser uniformes, no qual os votos dos classificadores possuem o mesmo peso, ou os votos podem possuir pesos, onde o voto de cada classificador possui um peso associado, podendo ser baseado em seu desempenho individual.

Para minimizar o *overfitting*, um conjunto de *holdout* pode ser utilizado para a avaliação do preditor, essa técnica envolve dividir o conjunto de dados disponível em dois subconjuntos distintos: um para treinamento e outro para teste. O conjunto de teste não é utilizado em nenhum momento para treinamento dos modelos e é apresentado ao modelo final para a validação de dados novos.

Seria interessante criar uma tabela para comparação de resultados com os trabalhos relacionados, com o intuito de facilitar a visualização dos resultados.

REFERÊNCIAS

- ABAYARATNE, S.; ILMINI, K.; FERNANDO, T. Identification of snake species found in sri lanka using convolutional neural networks. In: . [S.l.: s.n.], 2019.
- ABDURRAZAQ, I. S.; SUYANTO, S.; UTAMA, D. Q. Image-based classification of snake species using convolutional neural network. In: **2019 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)**. [S.l.: s.n.], 2019. p. 97–102.
- AHMED, K.; GAD, M. A.; ABOUTABL, A. E. Snake species classification using deep learning techniques. **Multimedia Tools and Applications**, Springer, 2023.
- ALLUGUNTI, V. R. A machine learning model for skin disease classification using convolution neural network. **International Journal of Computing, Programming and Database Management**, v. 3, n. 1, p. 141–147, 2022.
- AMIR, A. et al. Image classification for snake species using machine learning techniques. In: PHON-AMNUAISUK, S.; AU, T.-W.; OMAR, S. (Ed.). **Computational Intelligence in Information Systems**. Cham: Springer International Publishing, 2017. p. 52–59. ISBN 978-3-319-48517-1.
- BORWARNGINN, P. et al. Knowing your dog breed: Identifying a dog breed with deep learning. **International Journal of Automation and Computing**, Springer, v. 18, p. 45–54, 2021.
- DANDENIYA, D.; WICKRAMASINGHE, B.; DASANAYAKA, C. A web-based application for snake species identification using vision transformer and cnn-based ensemble meta classifier. In: **2022 IEEE Pune Section International Conference (PuneCon)**. [S.l.: s.n.], 2022. p. 1–5.
- DUBE, S. S.; BHURU, A. Snake identification system using convolutional neural networks. In: IEEE. **2022 1st Zimbabwe Conference of Information and Communication Technologies (ZCICT)**. [S.l.], 2022. p. 1–5.
- DUCHI, J.; HAZAN, E.; SINGER, Y. Adaptive subgradient methods for online learning and stochastic optimization. **Journal of machine learning research**, v. 12, n. 7, 2011.
- DURSO, A. M. et al. Crowdsourcing snake identification with online communities of professional herpetologists and avocational snake enthusiasts. **Royal Society Open Science**, The Royal Society, v. 8, n. 1, p. 201273, 2021. Available from Internet: <<https://doi.org/10.1098/rsos.201273>>.
- FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2021.
- GAUS, Y. F. A. et al. Evaluating the transferability and adversarial discrimination of convolutional neural networks for threat object detection and classification within x-ray security imagery. In: **2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)**. [S.l.: s.n.], 2019. p. 420–425.

GBIF: The Global Biodiversity Information Facility. **What is GBIF?** 2024. Available from Internet: <<https://www.gbif.org/what-is-gbif>>.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [S.l.]: MIT Press, 2016. <<http://www.deeplearningbook.org>>.

GUEDES, T. B.; ENTIAUSPE-NETO, O. M.; COSTA, H. C. **Lista de répteis do Brasil: atualização de 2022**. Zenodo, 2023. Available from Internet: <<https://doi.org/10.5281/zenodo.7829013>>.

HANGARAGI, S.; SINGH, T.; N, N. Face detection and recognition using face mesh and deep neural network. **Procedia Computer Science**, v. 218, p. 741–749, 2023. ISSN 1877-0509. International Conference on Machine Learning and Data Engineering. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S1877050923000546>>.

HE, K. et al. Deep residual learning for image recognition. **CoRR**, abs/1512.03385, 2015. Available from Internet: <<http://arxiv.org/abs/1512.03385>>.

HE, K. et al. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016. p. 770–778.

HUANG, G. et al. Densely connected convolutional networks. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2017.

HUSSAIN, M.; BIRD, J. J.; FARIA, D. R. A study on cnn transfer learning for image classification. In: SPRINGER. **Advances in Computational Intelligence Systems: Contributions Presented at the 18th UK Workshop on Computational Intelligence, September 5-7, 2018, Nottingham, UK**. [S.l.], 2019. p. 191–202.

KALINATHAN, L. et al. Automatic snake classification using deep learning algorithm. In: **CLEF (Working Notes)**. [S.l.: s.n.], 2021. p. 1587–1596.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv preprint arXiv:1412.6980**, 2014.

LEE, E.; KIM, D. Accurate traffic light detection using deep neural network with focal regression loss. **Image and Vision Computing**, v. 87, p. 24–36, 2019. ISSN 0262-8856. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S0262885619300538>>.

LI, Z. et al. A survey of convolutional neural networks: Analysis, applications, and prospects. **IEEE Transactions on Neural Networks and Learning Systems**, v. 33, n. 12, p. 6999–7019, 2022.

MINISTÉRIO DA SAÚDE. **Boletim Epidemiológico Vol. 53 - nº 36**. 2022. <<https://www.gov.br/saude/pt-br/centrais-de-conteudo/publicacoes/boletins/epidemiologicos/edicoes/2022/boletim-epidemiologico-vol-53-no36/view>>. Acesso em: 15 jan. 2024.

MINISTÉRIO DA SAÚDE. **Acidentes Ofídicos**. s.d. <<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/a/animais-peconhentos/acidentes-ofidicos>>. Acesso em: 14 jan. 2024.

MUMUNI, A.; MUMUNI, F. Data augmentation: A comprehensive survey of modern approaches. **Array**, v. 16, p. 100258, 2022. ISSN 2590-0056. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S2590005622000911>>.

OSSIBOFF, R. J. Chapter 37 - serpentes. In: TERIO, K. A.; MCALOOSE, D.; LEGER, J. S. (Ed.). **Pathology of Wildlife and Zoo Animals**. Academic Press, 2018. p. 897–919. ISBN 978-0-12-805306-5. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/B9780128053065000377>>.

OTHMAN, Z. et al. Snake species identification using digital image processing. In: **2021 6th IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE)**. [S.l.: s.n.], 2021. v. 6, p. 1–6.

PHUNG; RHEE. A high-accuracy model average ensemble of convolutional neural networks for classification of cloud image patches on small datasets. **Applied Sciences**, v. 9, p. 4500, 10 2019.

PROGGA, N. I. et al. A cnn based model for venomous and non-venomous snake classification. In: MAHMUD, M. et al. (Ed.). **Applied Intelligence and Informatics**. Cham: Springer International Publishing, 2021. p. 216–231. ISBN 978-3-030-82269-9.

RABOSKY, D. et al. Coral snakes predict the evolution of mimicry across new world snakes. **Nature Communications**, v. 7, p. 11484, 2016. Available from Internet: <<https://doi.org/10.1038/ncomms11484>>.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.

SAPONARA, S.; ELHANASHI, A. Impact of image resizing on deep learning detectors for training time and model performance. In: SPRINGER. **International Conference on Applications in Electronics Pervading Industry, Environment and Society**. [S.l.], 2021. p. 10–17.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. 2015.

SUTSKEVER, I. et al. On the importance of initialization and momentum in deep learning. In: PMLR. **International conference on machine learning**. [S.l.], 2013. p. 1139–1147.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. [S.l.: s.n.], 2016.

THARWAT, A. Classification assessment methods. **Applied computing and informatics**, Emerald Publishing Limited, v. 17, n. 1, p. 168–192, 2020.

TIELEMAN, T.; HINTON, G. et al. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. **COURSERA: Neural networks for machine learning**, v. 4, n. 2, p. 26–31, 2012.

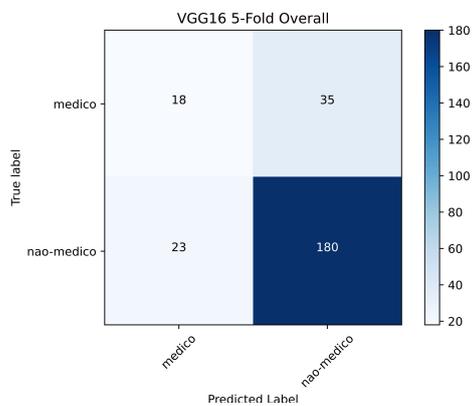
VASMATKAR, M. et al. Snake species identification and recognition. In: IEEE. **2020 IEEE Bombay Section Signature Conference (IBSSC)**. [S.l.], 2020. p. 1–5.

YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, v. 415, p. 295–316, 2020. ISSN 0925-2312. Available from Internet: <<https://www.sciencedirect.com/science/article/pii/S0925231220311693>>.

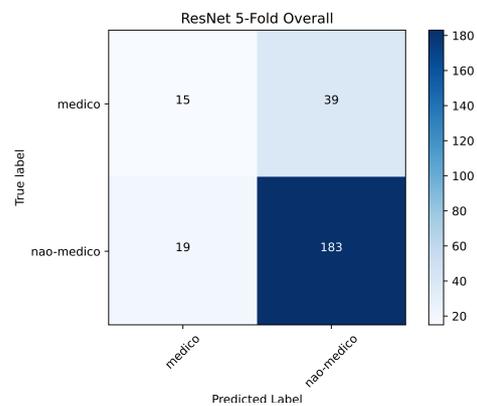
YOSINSKI, J. et al. **How transferable are features in deep neural networks?** 2014.

APÊNDICE A — MATRIZES DE CONFUSÃO DAS ARQUITETURAS TESTADAS

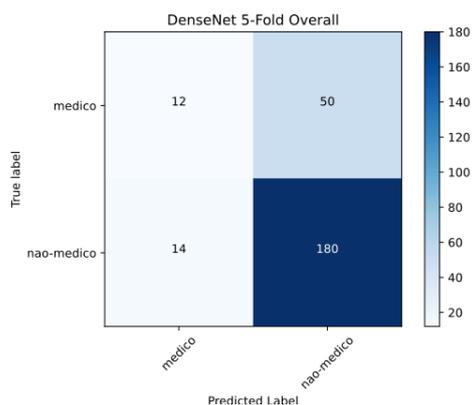
Figura A.1 – *Batch size* de 256 e 20 épocas de treinamento.



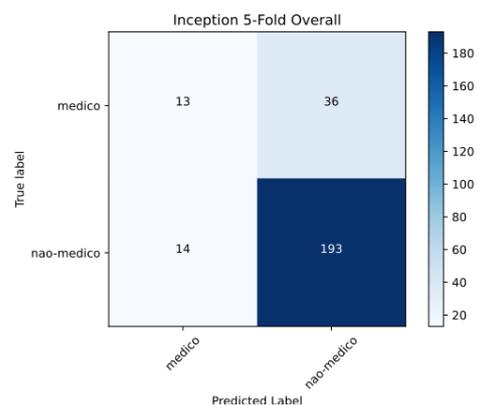
(a) Matriz de confusão da validação cruzada na VGG.



(b) Matriz de confusão da validação cruzada na ResNet.



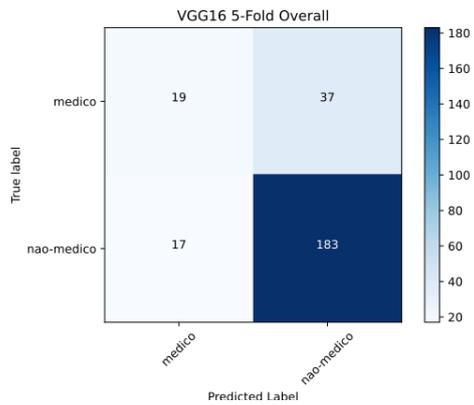
(c) Matriz de confusão da validação cruzada na DenseNet.



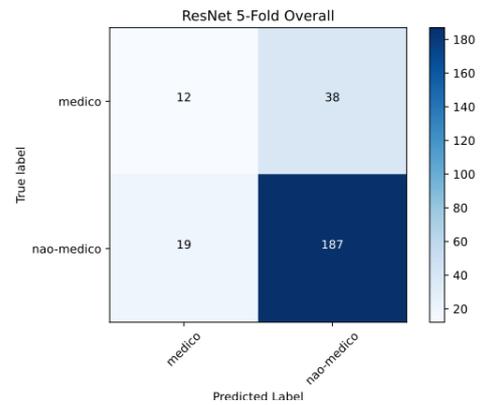
(d) Matriz de confusão da validação cruzada na Inception.

Fonte: O Autor.

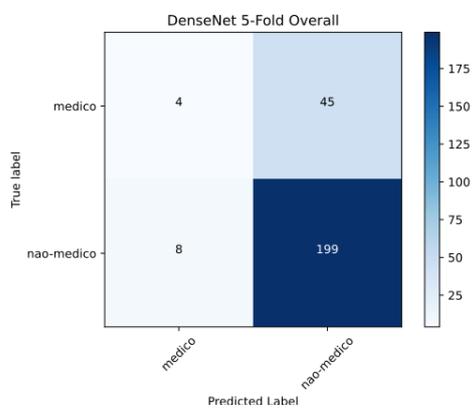
Figura A.2 – *Batch size* de 256 e 35 épocas de treinamento.



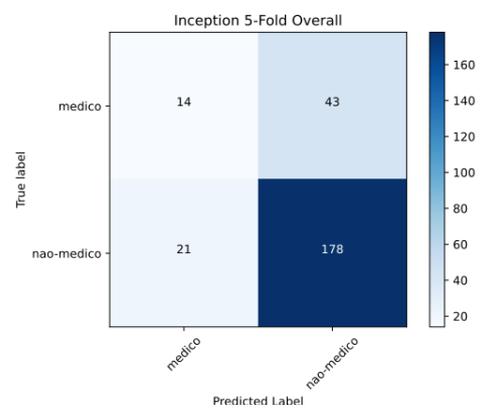
(a) Matriz de confusão da validação cruzada na VGG.



(b) Matriz de confusão da validação cruzada na ResNet.



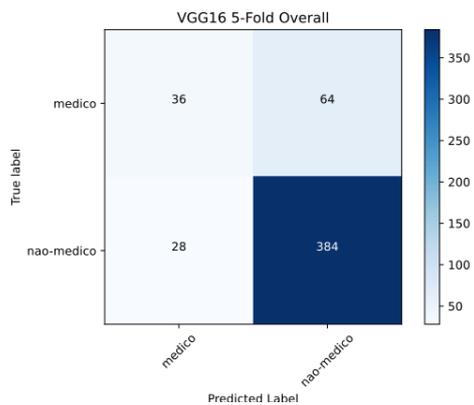
(c) Matriz de confusão da validação cruzada na DenseNet.



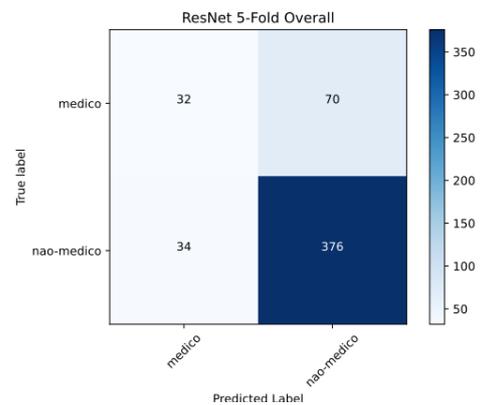
(d) Matriz de confusão da validação cruzada na Inception.

Fonte: O Autor.

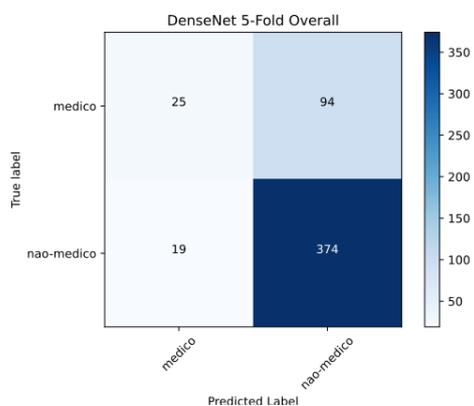
Figura A.3 – *Batch size* de 512 e 20 épocas de treinamento.



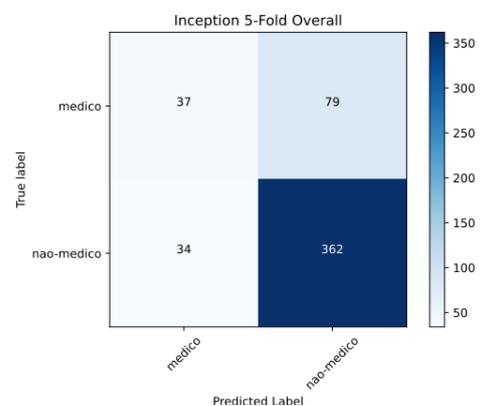
(a) Matriz de confusão da validação cruzada na VGG.



(b) Matriz de confusão da validação cruzada na ResNet.



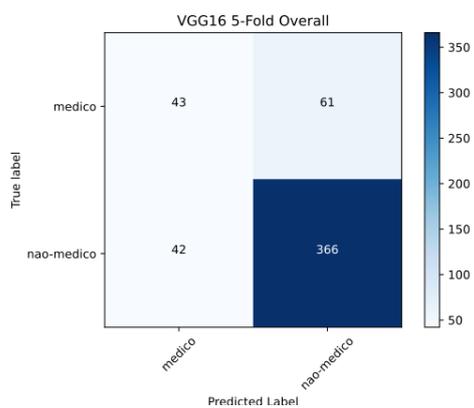
(c) Matriz de confusão da validação cruzada na DenseNet.



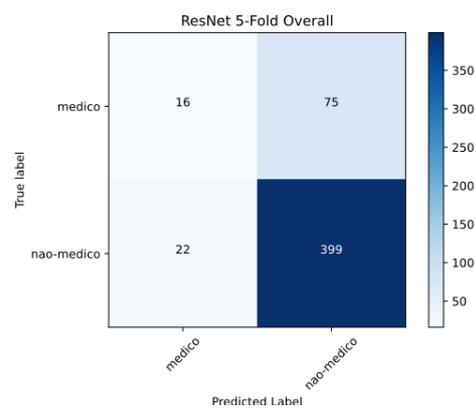
(d) Matriz de confusão da validação cruzada na Inception.

Fonte: O Autor.

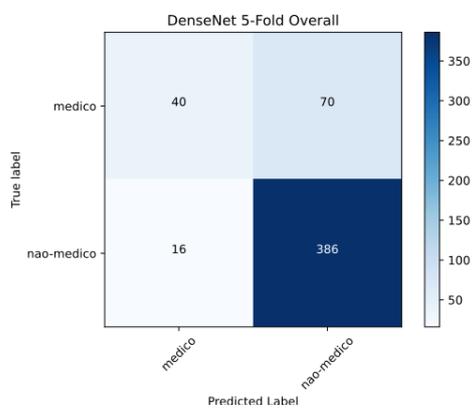
Figura A.4 – *Batch size* de 512 e 35 épocas de treinamento.



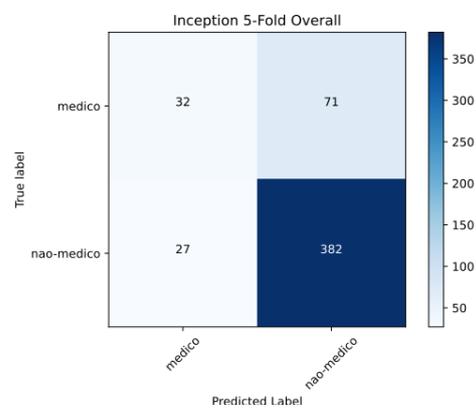
(a) Matriz de confusão da validação cruzada na VGG.



(b) Matriz de confusão da validação cruzada na ResNet.



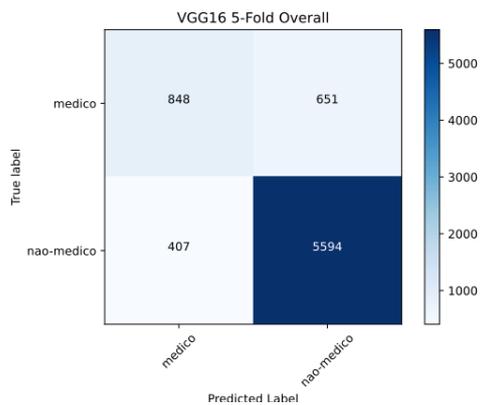
(c) Matriz de confusão da validação cruzada na DenseNet.



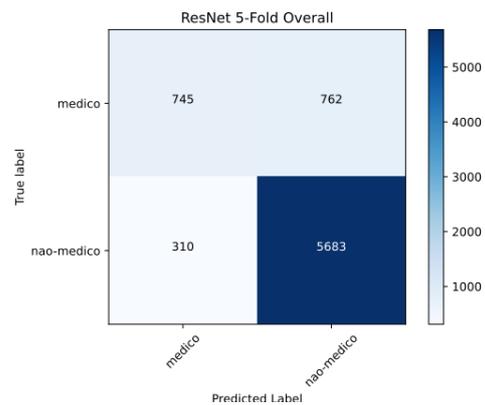
(d) Matriz de confusão da validação cruzada na Inception.

Fonte: O Autor.

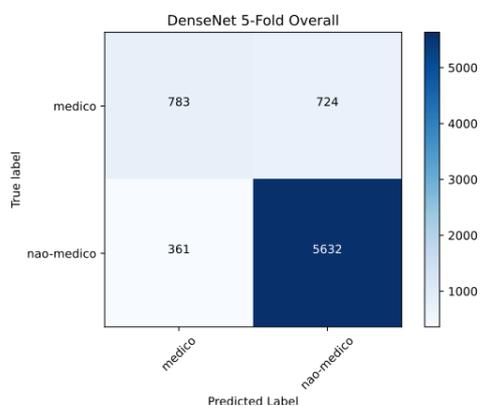
Figura A.5 – *Batch size* de 7500 e 20 épocas de treinamento.



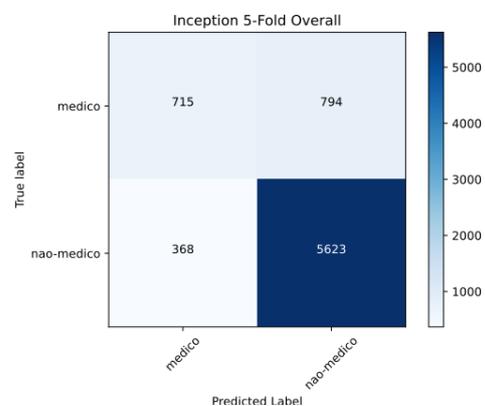
(a) Matriz de confusão da validação cruzada na VGG.



(b) Matriz de confusão da validação cruzada na ResNet.



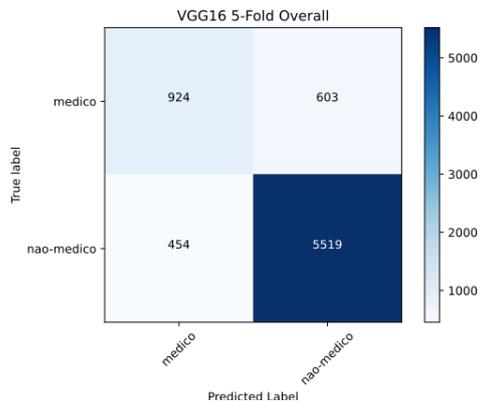
(c) Matriz de confusão da validação cruzada na DenseNet.



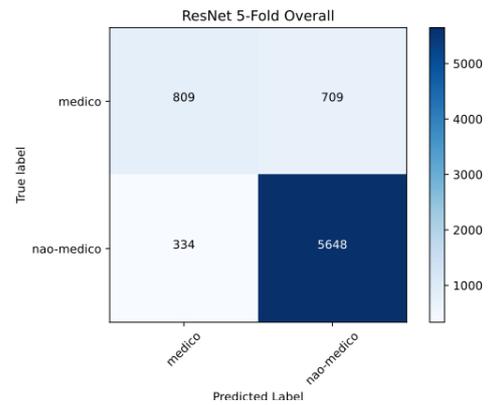
(d) Matriz de confusão da validação cruzada na Inception.

Fonte: O Autor.

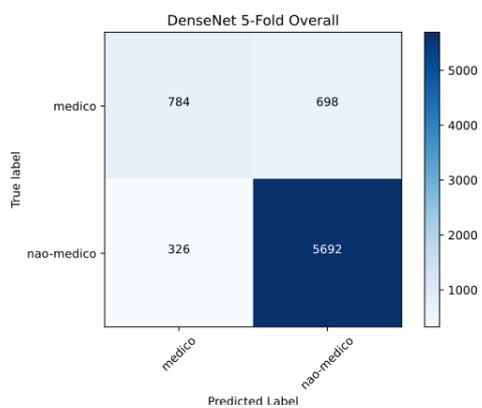
Figura A.6 – *Batch size* de 7500 e 35 épocas de treinamento.



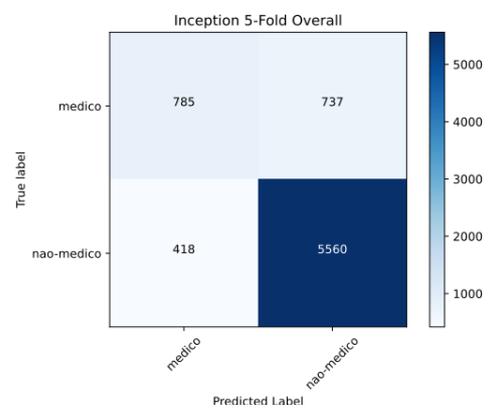
(a) Matriz de confusão da validação cruzada na VGG.



(b) Matriz de confusão da validação cruzada na ResNet.



(c) Matriz de confusão da validação cruzada na DenseNet.



(d) Matriz de confusão da validação cruzada na Inception.

Fonte: O Autor.