



# Análise Comparativa de Redes Neurais em Classificação de Imagens de Animais

*G. P. Barros*      *H. Pedrini*

Relatório Técnico - IC-PFG-23-35  
Projeto Final de Graduação  
2023 - Dezembro

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

The contents of this report are the sole responsibility of the authors.  
O conteúdo deste relatório é de única responsabilidade dos autores.

# Análise Comparativa de Redes Neurais em Classificação de Imagens de Animais

Gustavo Praciano Barros\*

Hélio Pedrini†

## Resumo

Este relatório de projeto final de graduação visa investigar o impacto de redes neurais conhecidas para um problema de classificação de imagens. Quatro redes neurais foram escolhidas para serem avaliadas em um problema de classificação de animais. A partir dessas duas escolhas iniciais, que serão descritas em detalhes no relatório, apresentaremos os métodos desenvolvidos e as bases de dados utilizadas, discutiremos os resultados experimentais encontrados focando em uma análise comparativa de redes neurais no contexto de classificação de imagens de animais.

## 1 Introdução

Aprendizado de máquina tem crescido significativamente em ciência da computação [16], o que cada vez mais vemos suas aplicações sendo empregadas e influenciando o mundo. Devido a isso, encontramos muitas soluções com métodos diferentes sendo aplicados para diversos contextos [4].

Devido a essa pluralidade, para um cientista da computação pode ser atualmente desafiador definir qual método usar para uma aplicação. Para um mesmo problema, podemos ter diferentes abordagens, não existindo uma solução que é melhor para todos os casos e cujos contextos têm diversas variáveis que impactam na escolha de uma abordagem [11]. Neste projeto, buscamos explorar abordagens conhecidas para analisar como métodos já bem estudados podem impactar o problema.

Um problema conhecido na área de aprendizado de máquina é o de classificação que, em sua essência, é buscar classificar determinado objeto de estudo em alguma classe previamente determinada a partir de suas características [4]. Além disso, classificação por si só pode ser explorada de várias maneiras [12] e, neste projeto, trabalharemos com classificação de imagens por meio de redes neurais. Somando-se a isso, escolhemos explorar classificações de imagens de animais, os quais possuem muitas espécies e várias classes biológicas associadas [18].

A principal motivação deste projeto é comparar resultados de redes neurais no contexto de classificação de imagens de animais e investigar os impactos das redes a partir da análise comparativa. Além disso, temos como objetivo apresentar um método que melhore a

---

\*Instituto de Computação, Universidade Estadual de Campinas, 13083-852, Campinas, SP.

†Instituto de Computação, Universidade Estadual de Campinas, 13083-852, Campinas, SP.

eficácia das redes utilizadas e diminua outros impactos para termos uma comparação mais relacionada às variantes desenvolvidas. Por fim, iremos explorar o desempenho de redes neurais para diferentes bases de dados de animais.

O texto está organizado como segue. A Seção 2 descreve os conceitos relacionados e tópicos relevantes para o entendimento do projeto. A Seção 3 apresenta os principais desafios intrínsecos às características do problema abordado. A Seção 4 descreve a metodologia usada neste trabalho. A Seção 5 descreve e analisa os resultados experimentais gerados a partir da classificação de imagens de animais. A Seção 6 apresenta as conclusões do trabalho e algumas propostas de trabalhos futuros.

## 2 Conceitos Relacionados

Antes de iniciarmos a propriamente descrever as principais etapas do método desenvolvido no projeto, alguns conceitos e temas importantes serão descritos para facilitar o entendimento do trabalho. Vale ressaltar que ideia é apresentar brevemente os temas, mais para o leitor adquirir algum embasamento para continuarmos a discussão do projeto.

### 2.1 Visão Computacional

A visão computacional é uma área da computação que busca capacitar os computadores a interpretar e compreender informações visuais do mundo real [19]. Utilizando algoritmos e técnicas de processamento de imagem, a visão computacional permite que os sistemas analisem e extraiam significado de dados visuais, como fotografias ou vídeos.

Ela abrange uma variedade de tarefas, tais como reconhecimento de padrões, detecção de objetos, rastreamento de movimento e reconhecimento facial. Essa disciplina tem aplicações em diversas áreas, incluindo medicina, automação industrial, veículos autônomos, biologia, segurança, entre outras, contribuindo para o avanço da interação entre máquinas e o ambiente visual ao nosso redor [20, 23].

Um exemplo de aplicação da visão computacional é o reconhecimento de caracteres em documentos digitalizados. Nesse contexto, algoritmos são empregados para analisar padrões visuais em imagens de texto, identificando letras e números de maneira automatizada [24]. Uma base de dados conhecida é o MNIST (*Modified National Institute of Standards and Technology* [2], que é amplamente utilizada como ponto de partida na área de visão computacional e problemas de classificação de imagens. Ela consiste em um conjunto de 70.000 imagens em escala de cinza de dígitos de 0 a 9 escritos à mão, cada uma com dimensões de  $28 \times 28$  pixels.

No contexto deste projeto, a classificação de imagens de animais é uma tarefa típica na área de visão computacional devido à necessidade de interpretar e compreender informações visuais complexas [6]. Esta área busca capacitar os computadores a extrair significado de dados visuais, sendo que a classificação de imagens de animais representa um desafio importante. A diversidade de características visuais dos animais, como formas, texturas e cores, chama a atenção dos estudiosos dessa área buscando explorar essas características de forma eficaz.

A aplicação da classificação de imagens de animais possui diversas utilidades práticas, desde a identificação de animais selvagens em estudos de conservação até sistemas de monitoramento de animais de estimação [15, 22]. A classificação de imagens de animais destaca a versatilidade e a importância da visão computacional em lidar com dados visuais diversificados em uma variedade de setores.

## 2.2 Redes Neurais

Redes neurais são modelos computacionais inspirados no funcionamento do sistema nervoso humano [8], projetadas para realizar tarefas de aprendizado e reconhecimento de padrões. Compostas por unidades interconectadas chamadas neurônios artificiais, essas redes formam camadas que processam dados de entrada, atribuem pesos às conexões entre neurônios e produzem saídas.

O aprendizado ocorre ajustando esses pesos com base em exemplos de treinamento, permitindo que a rede generalize e responda a novos dados. Redes neurais são amplamente utilizadas em diversas áreas, como o processamento de linguagem natural, a tomada de decisões em tarefas complexas e, para o caso deste projeto, o reconhecimento de imagens, contribuindo para avanços significativos em inteligência artificial.

As redes neurais artificiais foram concebidas com base na estrutura e no funcionamento das redes neurais biológicas presentes no cérebro humano. Cada neurônio artificial em uma rede neural computacional corresponde, em certo sentido, a um neurônio biológico, e as conexões entre eles refletem as sinapses que permitem a comunicação entre os neurônios. O processo de aprendizado em redes neurais computacionais, conhecido como treinamento, é inspirado no modo como o cérebro humano se adapta e aprende com a experiência.

Essa abordagem busca imitar a plasticidade cerebral e a capacidade do sistema nervoso para reconhecimento de padrões, permitindo que as redes neurais computacionais realizem tarefas complexas e aprendam a partir de dados. Essa convergência entre a ciência da computação e a neurociência tem sido fundamental para impulsionar os avanços na inteligência artificial, aproximando a capacidade das máquinas da complexidade e flexibilidade do cérebro humano [14].

Redes neurais computacionais são organizadas em camadas, incluindo uma camada de entrada para receber dados, camadas ocultas para processamento e extração de características e uma camada de saída para produzir resultados. Cada camada opera por meio de conexões ponderadas entre neurônios, permitindo que a rede aprenda automaticamente a hierarquia de características nos dados durante o treinamento. Essa estrutura em camadas é fundamental para lidar com problemas complexos de aprendizado, facilitando a adaptação a padrões e a execução de tarefas específicas.

Em nosso projeto, quatro arquiteturas de redes neurais convolucionais serão exploradas em tarefas de visão computacional: AlexNet, ResNet, VGG16 e EfficientNet. Cada uma dessas arquiteturas contribuiu significativamente para o avanço da visão computacional [3] ao longo dos últimos anos, com escolhas entre elas guiadas por requisitos específicos de aplicação e recursos computacionais disponíveis. Devido a isso, escolhemos essas como as redes neurais para abordar o problema de classificação de imagens neste projeto.

AlexNet [3], uma das primeiras redes neurais convolucionais a ganhar destaque, introdu-

ziu a ideia de treinamento eficiente usando várias placas gráficas (*graphics processing units* - GPUs). Com cinco camadas convolucionais e três camadas totalmente conectadas, ela obteve sucesso em competições de reconhecimento de imagens.

ResNet [3], ou Redes Residuais, inovou ao introduzir blocos residuais, incorporando atalhos para facilitar o fluxo de informação durante o treinamento. Essa abordagem permitiu a construção de redes mais profundas, resolvendo o problema do desaparecimento do gradiente e possibilitando a efetiva utilização de redes com centenas de camadas.

VGG16 [3], por sua vez, é conhecida por sua arquitetura simples e uniforme, utilizando convoluções  $3 \times 3$  em toda a rede. Há várias versões com diferentes números de camadas e a ênfase na profundidade e na simplicidade caracteriza a abordagem da VGG16.

EfficientNet [10] introduziu uma perspectiva de equilíbrio entre largura, profundidade e resolução para otimizar a eficiência computacional e a precisão. Ao incorporar escalas diferentes para largura, profundidade e resolução, a EfficientNet foi projetada para alcançar alto desempenho em tarefas de visão computacional com um custo computacional reduzido. Além disso, as versões B0, B1, B2, B3, B4, B5, B6 e B7 da EfficientNet representam diferentes escalas da arquitetura, cada uma otimizada para um determinado nível de complexidade e tamanho de modelo. O índice de escala, de 0 a 7, indica o grau de expansão em largura, profundidade e resolução da rede.

## 2.3 Treinamento de Modelos

Um fluxo clássico de treinamento de modelos para classificação em aprendizado de máquina inicia-se com a coleta e preparação dos dados [4], que envolve a aquisição de conjuntos de treinamento e teste, além de pré-processamento para normalização e tratamento de valores ausentes. Em seguida, os dados são divididos em conjuntos de treinamento e validação.

A separação de conjuntos de treinamento e validação é crucial no aprendizado de máquina porque ajuda a avaliar o desempenho do modelo de forma independente em dados não vistos durante o treinamento [17]. Isso previne o sobreajuste (*overfitting*), em que o modelo se adapta demais aos dados de treinamento e falha em generalizar para novos dados. Essa prática promove a criação de modelos mais robustos e confiáveis, auxiliando que o desempenho observado durante o treinamento seja um reflexo mais preciso da capacidade do modelo em lidar com dados do mundo real.

Além disso, o conjunto de validação serve como um retorno no treinamento em tempo real e indicador imparcial da capacidade de generalização do modelo, permitindo ajustes de hiperparâmetros e arquitetura sem comprometer a avaliação com os dados de treinamento. Hiperparâmetros são configurações externas ao modelo de aprendizado de máquina que afetam seu treinamento e desempenho, como taxa de aprendizado e número de camadas em uma rede neural. Eles são definidos antes do treinamento geralmente pelo programador e influenciam a eficácia do modelo.

Após os preparativos e o pré processamento dos dados, a escolha e a configuração do modelo são etapas fundamentais, devido a cada modelo ter suas vantagens e desvantagens que se encaixam melhor para determinados problemas. Portanto, explorar essas diferenças é um dos objetivos do nosso projeto.

Com a seleção de algoritmos e ajuste de hiperparâmetros para otimizar o desempenho, o treinamento do modelo ocorre utilizando o conjunto de treinamento, e a validação é realizada no conjunto de validação para avaliar a eficácia.

Após o treinamento satisfatório, o modelo é testado no conjunto de teste para avaliar sua generalização para dados não vistos. A análise dos resultados, ajustes finos e, eventualmente, a implementação do modelo em ambiente de produção encerram um fluxo clássico de treinamento, com a possibilidade de iterações para melhorias contínuas.

De modo geral, este processo descrito forma a base para a construção, avaliação e implementação bem-sucedida de modelos de classificação em aprendizado de máquina. Esse processo por mais que possa ser explicado de maneira simples em um fluxo, o treinamento de modelos possui complexidade e variedades em cada etapa que o compõe.

Por exemplo, a escolha de hiperparâmetros, como taxa de aprendizado, número de camadas e tamanho do lote, interagem de maneiras complexas e sutis, impactando a capacidade do modelo de aprender padrões representativos nos dados [17]. Encontrar a combinação ideal desses hiperparâmetros muitas vezes requer uma exploração extensiva, pois não existe uma abordagem única que se aplique universalmente. A seleção cuidadosa é fundamental para evitar *overfitting* ou *underfitting*, que ocorre quando um modelo é muito simples para representar os padrões nos dados, resultando em desempenho insatisfatório, e a complexidade é aumentada pela necessidade de equilibrar o desempenho desejado com os recursos computacionais disponíveis.

Portanto, a escolha de hiperparâmetros é um desafio comum no desenvolvimento de modelos de classificação de imagem, exigindo experimentação iterativa e compreensão profunda do problema específico em questão. Vale ressaltar que a escolha de hiperparâmetros é responsabilidade do pesquisador, o que muitas vezes não temos um método ou indicação de parâmetros bons para começar um treinamento.

Na etapa final do treinamento, usamos o conjunto de dados de teste para avaliarmos o desempenho do modelo para dados não vistos no treinamento. Usamos métricas para avaliarmos, estas tem diversas com cálculos e objetivos diferentes de avaliação [25]. Para o nosso projeto, empregaremos as métricas acurácia, acurácia balanceada, medida F1, revocação e precisão.

A acurácia, comumente utilizada em configurações de aprendizado de máquina, reflete a precisão global do modelo, sendo calculada como a soma dos verdadeiros positivos e verdadeiros negativos dividida pelo total de exemplos, especialmente em cenários de classificação binária. Contrariamente, a acurácia balanceada, também denominada acurácia média, oferece uma abordagem mais equitativa em situações de desbalanceamento entre classes. Calculada como a média das acurácias individuais por classe, proporciona uma avaliação imparcial do desempenho do modelo, especialmente relevante quando se busca evitar viés em direção à classe majoritária. Este conceito contribui para uma análise mais abrangente da eficácia do modelo, por exemplo, em nosso contexto no qual as bases de dados possuem distribuição heterogênea de classes.

A revocação, também chamada de sensibilidade, reflete a capacidade de um modelo identificar corretamente todos os exemplos positivos em um conjunto de dados. É calculada como a proporção de verdadeiros positivos em relação à soma de verdadeiros positivos e falsos negativos. Essa métrica é crucial em cenários onde a detecção de todos os casos posi-

tivos é essencial, como em diagnósticos médicos ou questões de segurança. Uma revocação alta indica uma boa capacidade do modelo em capturar a maioria dos exemplos positivos, enquanto uma revocação baixa sugere que muitos casos positivos estão sendo negligenciados durante a classificação.

A métrica de precisão em aprendizado de máquina avalia a habilidade de um modelo em identificar corretamente exemplos positivos em relação ao total de exemplos classificados como positivos. Calculada como a proporção de verdadeiros positivos sobre a soma de verdadeiros positivos e falsos positivos, a precisão fornece uma visão crítica da confiabilidade das classificações positivas do modelo. Essa métrica é importante em cenários médicos ou de segurança, em que a minimização de falsos positivos é essencial, contribuindo para uma avaliação abrangente da eficácia do modelo.

A medida F1, um indicador consolidado no âmbito do aprendizado de máquina, combina as métricas de precisão e revocação para fornecer uma avaliação equilibrada do desempenho de um modelo, especialmente relevante em cenários de desbalanceamento entre classes. Calculada como a média harmônica entre precisão e revocação, a medida F1 visa mitigar a influência de resultados desproporcionais e oferecer uma métrica única que pondera tanto falsos positivos quanto falsos negativos. Sua aplicação é fundamental em contextos nos quais a busca por um equilíbrio entre a identificação correta de exemplos positivos e a minimização de falsas identificações é crítica, aprimorando assim a avaliação da eficácia do modelo em tarefas de classificação.

Em contexto de multiclasse, a renovação, precisão e medida F1 possuem calculados diferentes, os quais são macro, micro e ponderada. O Macro, ao calcular a média para cada classe, atribui igual peso a todas as classes, independentemente de sua frequência no conjunto de dados. Em contraste, o Micro agrega os verdadeiros positivos, falsos positivos e falsos negativos de todas as classes antes de calcular a medida, tratando cada instância de maneira uniforme e sendo sensível a classes mais frequentes. Já o ponderado, ao calcular para cada classe e aplicar uma média ponderada proporcional à presença das classes no conjunto de dados, equilibra a importância relativa e a frequência das classes. A escolha entre essas abordagens depende das nuances específicas do problema e das prioridades na avaliação do desempenho do modelo em tarefas de classificação multiclasse.

### 3 Desafios

Nesta seção, alguns desafios previstos para alcançar os nossos objetivos são descritos, alguns mapeados inicialmente e outros que foram sendo encontrados durante o desenvolvimento do projeto.

A escolha de hiperparâmetros no treinamento de redes neurais [17] é um desafio devido à complexidade e à interdependência desses parâmetros em influenciar o desempenho do modelo. Hiperparâmetros, como taxa de aprendizado, tamanho do lote e arquitetura da rede, não são aprendidos durante o treinamento, tornando-se decisões críticas antes do início do processo.

A mudança sutil entre esses parâmetros significa que ajustar um deles pode ter impacto direto em outros, tornando o processo de otimização uma tarefa desafiadora e, muitas

vezes, heurística. Além disso, o espaço de busca para encontrar a combinação ideal de hiperparâmetros é vasto, demandando recursos computacionais consideráveis.

Portanto, encontrar a configuração mais adequada requer uma abordagem cuidadosa, envolvendo experimentação iterativa e compreensão profunda do domínio do problema, tornando a escolha de hiperparâmetros um desafio constante e essencial no treinamento de redes neurais.

A análise comparativa de modelos de redes neurais para conjuntos de dados requer não apenas a compreensão das nuances intrínsecas aos modelos, mas também a habilidade de isolar e minimizar outras influências que podem distorcer os resultados [13, 1, 9].

Maximizar a visualização dos impactos das mudanças nas redes envolve não apenas a seleção cuidadosa de métricas de avaliação, mas também a criação de experimentos controlados. O desafio reside em garantir que as diferenças observadas sejam atribuíveis às variações nas arquiteturas ou hiperparâmetros das redes, e não a fatores externos, como distribuição desigual de dados, ruídos ou peculiaridades específicas do conjunto de dados.

Além disso, a consistência nos procedimentos de treinamento e avaliação é crucial para garantir resultados confiáveis e generalizáveis. Enfrentar esses desafios exige uma abordagem metodológica sólida, rigor na experimentação e uma compreensão do contexto específico do problema em análise.

Por fim, problemas de convergência, estes são frequentes devido à complexidade do espaço de otimização e à sensibilidade dos modelos a diferentes configurações [17]. A convergência ideal, onde o modelo atinge um estado estável e ótimo em termos de desempenho, pode ser prejudicada por diversos fatores. Taxas de aprendizado inadequadas, por exemplo, podem resultar em atualizações de parâmetros muito pequenas ou muito grandes, dificultando a convergência eficiente.

Além disso, a presença de platôs ou mínimos locais no espaço de otimização pode fazer com que o algoritmo de treinamento estagne antes de atingir a solução ótima. Problemas de inicialização de pesos, escolha inadequada de arquitetura e dados ruidosos também contribuem para desafios na convergência. Lidar com essas questões exige uma cuidadosa sintonia de hiperparâmetros, técnicas avançadas de inicialização e, em alguns casos, a aplicação de estratégias como normalização de lotes ou regularização para promover uma convergência mais estável e eficaz durante o treinamento.

## 4 Metodologia

A metodologia está dividida em quatro partes distintas: coleta de dados, pré-processamento, treinamento e escolha de resultados. Essa divisão proporcionará uma compreensão mais clara do nosso trabalho, delineando sistematicamente a metodologia empregada no projeto.

Adicionalmente, desenvolvemos um *script* abrangente que automatiza tanto os procedimentos de treinamento dos modelos quanto o acesso às bases de dados em uma única execução. A configuração prévia das bases de dados a serem utilizadas e comandos dos modelos a serem treinados é o único requisito para a eficácia desse código. Essa abordagem é importante no contexto do nosso fluxo de trabalho, pois o *script* segue meticulosamente



os princípios metodológicos estabelecidos.

#### 4.1 Coleta de Dados

Antes de iniciar o projeto, conduzimos uma pesquisa extensiva para identificar bases de dados que representassem de forma abrangente o nosso problema de classificação de animais. Com base na diversidade e na complexidade inerentes ao reino animal, optamos por focar em três categorias principais: animais terrestres, animais aquáticos e aves.

Com a especificação das categorias, diversas opções de bases de dados provenientes de fontes variadas foram consultadas. Para garantir a relevância e qualidade dos dados, decidimos restringir nossa pesquisa a bases de dados públicas, com o Kaggle sendo nossa principal fonte. Essa escolha foi motivada pela participação ativa da comunidade de ciência de dados nessa plataforma, garantindo atualizações regulares e contribuições de qualidade [5].

Além disso, optamos por escolher bases de dados públicas. Essa preferência traz consigo vantagens, como acessibilidade para a comunidade científica, transparência, bem como a possibilidade de replicação e validação dos resultados por outros pesquisadores. O uso de dados públicos também contribui para a reprodutibilidade da pesquisa, fortalecendo a integridade e confiabilidade dos resultados obtidos no projeto.

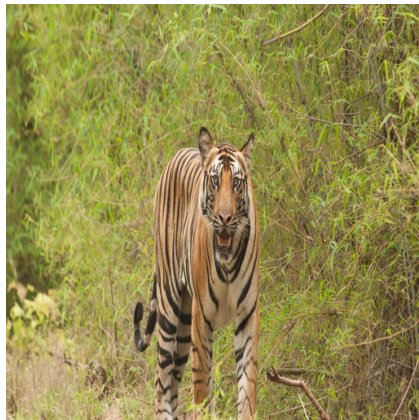
No Kaggle, priorizamos bases de dados com maior visibilidade, considerando visualizações, *downloads*, recomendações e contribuições de códigos. Este critério é valioso porque reflete o uso ativo dessas bases de dados por outros membros da comunidade. A alta visibilidade sugere que esses conjuntos de dados são amplamente reconhecidos e utilizados, indicando uma aceitação generalizada pela comunidade.

Além disso, a quantidade significativa de visualizações e *downloads* sugere um interesse substancial, enquanto as contribuições de códigos adicionam uma camada adicional de validação, demonstrando a utilidade prática desses conjuntos de dados em projetos semelhantes. Dessa forma, a combinação desses critérios reforça a robustez da nossa escolha e a confiabilidade dos dados selecionados para o desenvolvimento da comparação de modelos.

Seguindo essas prioridades, identificamos quatro conjuntos de dados que atendiam às nossas necessidades. O primeiro abrange animais terrestres, incluindo espécies populares, como leão e tigre. O segundo foca em répteis, uma classe que decidimos separar devido às suas características únicas, como escamas, ambiente de vida e padrões de cores. O terceiro engloba animais aquáticos, como golfinhos e polvos, e o quarto concentra-se em aves, oferecendo uma ampla variedade de espécies, desde as mais conhecidas até aquelas específicas de determinadas regiões. As Figuras 1(a), (b), (c) e (d) mostram, respectivamente, imagens dos conjuntos de dados de animais terrestres, répteis, aves e animais aquáticos.

#### 4.2 Pré-Processamento

Com as bases de dados escolhidas, implementamos um processo de pré-processamento para adequá-las efetivamente ao treinamento. Inicialmente, padronizamos as imagens para dimensões de  $224 \times 224 \times 3$ , alinhando com os requisitos de entrada das redes ResNet e VGG16. Para a EfficientNet, cujo tamanho pode variar, mantivemos a dimensão padrão estabelecida, pois a entrada típica para muitas implementações da EfficientNet é de  $224 \times 224$



(a) imagem de um tigre



(b) imagem de uma iguana



(c) Imagem de um azure jay



(d) Imagem de golfinhos

Figura 1: Imagens de animais dos conjuntos de dados.

pixels.

Para a AlexNet, utilizamos a função de redimensionamento do PyTorch [21] para ajustar as imagens para  $227 \times 227 \times 3$ , que seria o tamanho típico usado por essa rede. Notamos que, dentro das bases de dados, apenas um apresentava imagens com as dimensões pré-definidas, requerendo, portanto, o redimensionamento das imagens nos demais conjuntos.

Outra consideração importante na criação das bases de dados é o número significativo de classes em alguns conjuntos de imagens. A presença de múltiplas classes e poucas amostras nas bases de dados dificulta o treinamento [7]. Isso implica maior número de características discriminativas entre elas em conjunto com a tarefa de classificar mais classes, sendo necessário mais exemplos. Em resposta a isso, ajustamos algumas das bases de dados, criando versões com um número reduzido de classes para facilitar a convergência dos modelos ao problema proposto. Essas modificações visam melhorar o desempenho e a eficácia dos modelos no contexto do nosso projeto. A Tabela 1 descreve essas versões.

Finalmente, procedemos com a segmentação das bases de dados em conjuntos de trei-

Tabela 1: Identificadores (IDs) usados no código para se referir às bases de dados e descrição do conjunto de dados

Identificador	Descrição
02_01	Base de dados de animais aquáticos pré-processado, 7 classes, baleias, golfinhos, enguia, tubarões, focas e raias marinhas.
03_01	Base de dados de reptis e anfíbios pré-processado, 6 classes, cobra, camaleão, tartaruga, crocodilo, lagartixa e sapo.
03_02	Base de dados de reptis e anfíbios pré-processado, 5 classes escolhidos, similares de corpo, no caso, rabo, 4 patas e porte pequeno para médio, lagartixa, salamandra, lagarto, iguana e camaleão.
04_redimensionado	Base de dados de animais selvagens, 6 classes, hiena, leão, tigre, lobo, guepardo e raposa, usamos somente as imagens com o tamanho $224 \times 224 \times 3$ desta base de dados
05_01	Base de dados de aves com menos classes, 6 classes escolhidas, buscamos colocar espécies que têm características diferentes em comparação à outra, em que os nomes das espécies estão em inglês: AZURE JAY, BLACK NECKED STILT, BOBOLINK, CHUKAR PARTRIDGE, FLAME TANAGER, GLOSSY IBIS, WHITE EARED HUMMINGBIRD

namento, validação e teste. A distribuição adotada consistiu em alocar 70% das amostras para o conjunto de treinamento, reservando 15% para o conjunto de validação e outros 15% para o conjunto de teste, com exceção de alguns conjuntos, em que a proporção foi ajustada para 60/20/20.

Antes de iniciar o treinamento, estabelecemos essa divisão e a mantivemos constante para todos os modelos subsequentes. A adoção dessa abordagem foi intencional, visando assegurar que todos os modelos compartilhassem os mesmos conjuntos de treinamento, validação e teste. Essa uniformidade na divisão tem o propósito de proporcionar uma base homogênea para comparação de resultados, garantindo que as diferenças observadas sejam atribuíveis às características intrínsecas dos modelos, e não a variações nos conjuntos de dados.

### 4.3 Treinamento

Nesta etapa, empregamos o *script* desenvolvido para simplificar o processo de criação de modelos. O código é estruturado para permitir a especificação dos bancos de dados a serem utilizados na execução, assim como os modelos e hiperparâmetros a serem treinados, o arquivo denominado `config.py` é onde definimos essas especificações.

A configuração dos treinamentos a serem executados é armazenada em uma lista de comandos. Cada comando na lista contém as informações que definimos para o respectivo treinamento. Após a conclusão de um treinamento, o código automaticamente avança para

o próximo comando na lista, possibilitando o treinamento consecutivo de vários modelos.

Uma vez concluídos os treinamentos programados, o *script* registra os resultados em um arquivo CSV, incluindo não apenas os modelos treinados, mas também diversas métricas, como acurácia, medida F1, precisão e revocação. Essa abordagem proporciona uma organização dos dados resultantes, permitindo uma análise e facilidade de visualização do resultados obtidos.

O *script* desempenha um papel fundamental nesta fase, automatizando e simplificando a geração e avaliação de diversos modelos para que possamos analisá los. No entanto, mesmo com essa ferramenta, é necessário realizar escolhas cuidadosas dos hiperparâmetros e analisar os resultados do treinamento para identificar o modelo mais eficaz.

Dessa forma, o processo implementado para encontrar os modelos de qualidade iniciou com a seleção inicial de hiperparâmetros, alinhados com aqueles utilizados nos códigos que serviram de inspiração para a criação dos modelos. Este primeiro treinamento serve como ponto de partida para avaliar os resultados e o desempenho do modelo.

Em geral, o desempenho do primeiro modelo não atendia às expectativas, levando-nos a realizar ajustes nos hiperparâmetros em busca de melhores resultados. No início desse ajuste, alterações foram realizadas na taxa de aprendizado para avaliar melhorias. Ao identificar uma taxa de aprendizado que convergia de maneira melhor para resultados melhores, outros hiperparâmetros eram ajustados.

Em alguns casos, alterações mais conservadoras em determinados hiperparâmetros foram realizadas, com o objetivo de reduzir a complexidade e manter menos variáveis associadas. O ajuste no tamanho do lote foi efetuado principalmente para reduzir o tempo de execução e contornar limitações de recursos computacionais.

O processo de ajuste de hiperparâmetros prosseguiu até as modificações descritas não surtirem mais efeitos significativos. Optamos por ajustar o número de épocas por último, permitindo que o modelo se aprofundasse na convergência após observarmos resultados promissores com outros hiperparâmetros [4]. Vale ressaltar que, para o treinamento da EfficientNet, nessa alteração de números de épocas, tivemos que alterar o decaimento da taxa de aprendizado. Assim, para cada decaimento de taxa de aprendizado, houve mais épocas de treinamento.

Além disso, monitoramos a variação da métrica acurácia e a função de perda durante o treinamento. Quando estes dois valores oscilavam em torno de uma média de valores, ou seja, indicando falta de melhorias significativas [4], identificávamos esses conjuntos de hiperparâmetros como aqueles que convergiam em um contexto específico. Assim, separamos os resultados dessas execuções e, caso se desejasse efetuar outros experimentos, buscamos execuções de valores mais fora do escopo do contexto oscilatório.

É relevante ressaltar que, embora tenhamos seguido um fluxo pré-definido nesta etapa de treinamento, ocasionalmente exploramos abordagens diferentes para utilizar os hiperparâmetros. Testar e avaliar estratégias alternativas em relação ao fluxo de treinamento estabelecido, até mesmo que não pareçam convencionais, proporcionam sugestões valiosas na busca por valores de hiperparâmetros que reduzam o desempenho do modelo.

É importante ressaltar que o êxito de testes em uma base de dados não significa sucesso em outra. Em resumo, optamos por seguir um fluxo pré-definido para orientar o trabalho. No entanto, também conduzimos numerosos experimentos que variaram os parâmetros

estabelecidos, visando explorar diferentes abordagens no uso dos hiperparâmetros.

Por fim, como mencionado anteriormente, realizamos diversos treinamentos, cada um devidamente registrado em arquivos CSV contendo resultados e métricas, as quais serão discutidos na próxima seção, juntamente com os modelos gerados.

#### 4.4 Seleção de Modelos

Diante dos diversos resultados obtidos com diferentes conjuntos de parâmetros, optamos, em geral, por selecionar os modelos com maior acurácia para a comparação. Embora as demais métricas estejam disponíveis para comparação, optamos por não atribuir um peso significativo a essas métricas na escolha final, utilizando-as mais adequadas para análises comparativas.

Alguns modelos destinados a análises específicas não apresentaram uma convergência satisfatória, o que levou à exclusão destes. No entanto, em alguns casos, optamos por mantê-los na análise, mesmo com resultados insatisfatórios, a fim de ilustrar como aquele modelo não gerou bons resultados.

Por fim, o *script* mencionado armazena os modelos treinados, juntamente com um arquivo CSV contendo métricas de avaliação, como acurácia, acurácia balanceada, medida F1, revocação e precisão. Como mencionado, simplificamos a escolha de métricas, priorizando a acurácia como parâmetro principal, pois, de maneira geral, ela indica o desempenho no conjunto de teste, mesmo que parcialmente [25].

## 5 Experimentos

Como pode ser observado em nossos resultados apresentados nas Tabelas 2, 3, 4, 5 e 6, não obtivemos resultados satisfatórios, com exceção de algumas situações. Dessa forma, o problema deve ser aprimorado em uma versão futura do trabalho.

Tabela 2: Avaliações dos resultados dos treinamentos com a base de dados 02\_01.

Modelo	Acurácia	Acc Blc	F1 Macro	F1 Micro	F1 Ponderada
AlexNet	0,456	0,455	0,45	0,456	0,453
ResNet	0,392	0,381	0,378	0,391	0,386
VGG16	0,49	0,485	0,484	0,49	0,489
EfficientNet-b0	0,5	0,504	0,496	0,5	0,501
EfficientNet-b1	0,422	0,405	0,39	0,422	0,4
EfficientNet-b2	0,453	0,428	0,409	0,453	0,424
EfficientNet-b3	0,447	0,426	0,404	0,447	0,418
EfficientNet-b4	0,502	0,509	0,501	0,502	0,502
EfficientNet-b5	0,498	0,502	0,497	0,498	0,502
EfficientNet-b6	0,439	0,443	0,441	0,439	0,442
EfficientNet-b7	0,381	0,392	0,378	0,381	0,374

Tabela 3: Avaliações dos resultados dos treinamentos com a base de dados 03\_01.

<b>Modelo</b>	<b>Acurácia</b>	<b>Acc Blc</b>	<b>F1 Macro</b>	<b>F1 Micro</b>	<b>F1 Ponderada</b>
AlexNet	0,555	0,461	0,456	0,555	0,557
ResNet	0,491	0,335	0,333	0,491	0,469
VGG16	0,649	0,534	0,543	0,649	0,634
EfficientNet-b0	0,557	0,402	0,388	0,557	0,54
EfficientNet-b1	0,555	0,415	0,425	0,555	0,54
EfficientNet-b2	0,581	0,452	0,454	0,581	0,565
EfficientNet-b3	0,606	0,477	0,486	0,606	0,597
EfficientNet-b4	0,611	0,444	0,457	0,611	0,588
EfficientNet-b5	0,581	0,415	0,424	0,581	0,559
EfficientNet-b6	0,534	0,398	0,395	0,534	0,52
EfficientNet-b7	0,539	0,415	0,414	0,539	0,539

Tabela 4: Avaliações dos resultados dos treinamentos com a base de dados 03\_02.

<b>Modelo</b>	<b>Acurácia</b>	<b>Acc Blc</b>	<b>F1 Macro</b>	<b>F1 Micro</b>	<b>F1 Ponderada</b>
AlexNet	0,251	0,2	0,08	0,251	0,101
ResNet	0,251	0,2	0,08	0,251	0,101
VGG16	0,251	0,2	0,08	0,251	0,101
EfficientNet-b0	0,361	0,301	0,282	0,361	0,329

Tabela 5: Avaliações dos resultados dos treinamentos com a base de dados 04.redimensionada.

<b>Modelo</b>	<b>Acurácia</b>	<b>Acc Blc</b>	<b>F1 Macro</b>	<b>F1 Micro</b>	<b>F1 Ponderada.</b>
AlexNet	0,833	0,833	0,832	0,833	0,834
ResNet	0,696	0,698	0,697	0,696	0,697
VGG16	0,786	0,782	0,785	0,786	0,787
EfficientNet-b0	0,661	0,659	0,661	0,661	0,663
EfficientNet-b1	0,665	0,668	0,67	0,665	0,671
EfficientNet-b2	0,661	0,661	0,654	0,661	0,656
EfficientNet-b3	0,681	0,684	0,682	0,681	0,683
EfficientNet-b4	0,704	0,702	0,708	0,704	0,71
EfficientNet-b5	0,65	0,649	0,654	0,65	0,655
EfficientNet-b6	0,619	0,617	0,62	0,619	0,623
EfficientNet-b7	0,669	0,672	0,67	0,669	0,671

Durante a condução deste estudo, identificamos algumas limitações significativas que podem impactar a robustez dos resultados obtidos. Primeiramente, observamos uma lacuna

Tabela 6: Avaliações dos resultados dos treinamentos com a base de dados 05\_01.

Modelo	Acurácia	Acc Blc	F1 Macro	F1 Micro	F1 Ponderada
AlexNet	1	1	1	1	1
ResNet	0,971	0,971	0,971	0,971	0,971
VGG16	0,971	0,971	0,971	0,971	0,971
EfficientNet-b0	0,829	0,829	0,834	0,829	0,834
EfficientNet-b1	0,857	0,857	0,864	0,857	0,864
EfficientNet-b2	0,829	0,829	0,829	0,829	0,829
EfficientNet-b3	0,943	0,943	0,942	0,943	0,942
EfficientNet-b4	0,829	0,829	0,819	0,829	0,819
EfficientNet-b5	0,829	0,829	0,823	0,829	0,823
EfficientNet-b6	0,914	0,914	0,909	0,914	0,909
EfficientNet-b7	0,886	0,886	0,887	0,886	0,887

na utilização de métodos mais eficazes para a validação durante o treinamento dos modelos. Por exemplo, optamos por não incorporar técnicas como validação cruzada em nossa metodologia, uma técnica comumente usada em treinamento de modelos. A ausência dessa abordagem não permitiu a exploração mais extensiva de diferentes partições dos conjuntos de treinamento, validação e teste.

No que diz respeito às bases de dados específicas, notamos que, mesmo ao dedicarmos atenção à escolha criteriosa das mesmas, alguns conjuntos, como o de répteis e animais aquáticos, poderiam ter sido mais minuciosamente avaliados. Por exemplo, a presença de categorias amplas, como a classe “peixe” em animais aquáticos, pode ter comprometido a qualidade da classificação, pois o que rotulamos como peixes possui milhares de espécies que são bem distintas. Este aspecto se destaca em comparação com bases de dados de aves, onde cada classe representa uma espécie específica, cujos conjuntos são amplamente utilizados em competições, inclusive os resultados dessa base de dados são significativamente melhores.

Por fim, a escolha das condições iniciais e dos hiperparâmetros de treinamento é fundamental para o desempenho dos modelos. No entanto, reconhecemos que nossa abordagem foi limitada ao utilizar valores iniciais, sem empregar métodos específicos para a busca eficiente de bons hiperparâmetros. Essa lacuna em nossa metodologia destaca a necessidade de desenvolver estratégias mais robustas para a exploração sistemática dos hiperparâmetros, visando melhorar o desempenho dos modelos.

## 6 Conclusões e Trabalhos Futuros

Neste projeto, comparamos diferentes redes neurais para o problema de classificação de imagens de animais. Apesar de resultados não satisfatórios em vários cenários, os experimentos foram úteis para tirar algumas conclusões e buscar direções para melhorar a eficácia dos modelos.

Além disso, como um projeto final de graduação, este trabalho foi bastante enriquece-

dor para a ampliação do conhecimento nas áreas de análise de imagens e aprendizado de máquina.

Como propostas para trabalhos futuros, recomenda-se a aplicação de diferentes valores para os hiperparâmetros envolvidos nos modelos testados, o uso de bases de dados com maior quantidade de imagens e a experimentação com outros modelos de redes neurais.

## Referências

- [1] Loknath Sai Ambati and Omar El-Gayar. Human activity recognition: a comparison of machine learning approaches. *Journal of the Midwest Association for Information Systems*, 2021(1):4, 2021.
- [2] Alejandro Baldominos, Yago Saez, and Pedro Isasi. A survey of handwritten character recognition with mnist and emnist. *Applied Sciences*, 9(15):3169, 2019.
- [3] Dulari Bhatt, Chirag Patel, Hardik Talsania, Jigar Patel, Rasmika Vaghela, Sharnil Pandya, Kirit Modi, and Hemant Ghayvat. CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, 10(20):2470, 2021.
- [4] Jaime G Carbonell, Ryszard S Michalski, and Tom M Mitchell. An overview of machine learning. *Machine Learning*, 1983.
- [5] DataScientest. Kaggle : All You Need to Know about this Platform, 2023.
- [6] Gabriel S Ferrante, Felipe M Rodrigues, Fernando RH Andrade, Rudinei Goularte, and Rodolfo I Meneguette. Understanding the state of the Art in Animal detection and classification using computer vision technologies. In *IEEE International Conference on Big Data (Big Data)*, pages 3056–3065. Ieee, 2021.
- [7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [8] Neha Gupta. Artificial Neural Network. *Network and Complex Systems*, 3(1):24–28, 2013.
- [9] Brandon Heung, Hung Chak Ho, Jin Zhang, Anders Knudby, Chuck E Bulmer, and Margaret G Schmidt. An overview and comparison of machine-learning techniques for classification purposes in digital soil mapping. *Geoderma*, 265:62–77, 2016.
- [10] Van-Thanh Hoang and Kang-Hyun Jo. Practical analysis on architecture of Efficient-Net. In *14th International Conference on Human System Interaction*, pages 1–4. IEEE, 2021.
- [11] Jean Kaddour, Aengus Lynch, Qi Liu, Matt J Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems. *arXiv preprint arXiv:2206.15475*, 2022.



- [12] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26:159–190, 2006.
- [13] Wolfgang Kratsch, Jonas Manderscheid, Maximilian Röglinger, and Johannes Seyfried. Machine learning in business process monitoring: a comparison of deep learning and classical approaches used for outcome prediction. *Business & Information Systems Engineering*, 63:261–276, 2021.
- [14] Anders Krogh. What are artificial neural networks? *Nature Biotechnology*, 26(2):195–197, 2008.
- [15] Guoming Li, Yanbo Huang, Zhiqian Chen, Gary D Chesser, Joseph L Purswell, John Linhoss, and Yang Zhao. Practices and applications of convolutional neural network-based computer vision systems in animal farming: A review. *Sensors*, 21(4):1492, 2021.
- [16] Teresa Bernarda Ludermir. Inteligência Artificial e Aprendizado de Máquina: estado atual e tendências. *Estudos Avançados*, 35:85–94, 2021.
- [17] Tom Mitchell. *Learning, Machine*. McGraw Hill, 1997.
- [18] Xun Long Ng, Kian Eng Ong, Qichen Zheng, Yun Ni, Si Yong Yeo, and Jun Liu. Animal kingdom: A large and diverse dataset for animal behavior understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19023–19034, 2022.
- [19] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision*. Academic press, 2019.
- [20] Thierry Pun, Guido Gerig, and Osman Ratib. Image analysis and computer vision in medicine. *Computerized Medical Imaging and Graphics*, 18(2):85–96, 1994.
- [21] PyTorch. RESIZE. <https://pytorch.org/vision/main/generated/torchvision.transforms.Resize.html>.
- [22] Gyanendra K Verma and Pragya Gupta. Wild animal detection using deep convolutional neural network. In *2nd International Conference on Computer Vision & Image Processing*, volume 2, pages 327–338. Springer, 2018.
- [23] Athanasios Voulodimos, Nikolaos Doulamis, George Bebis, and Tania Stathaki. Recent developments in deep learning for engineering applications. *Computational Intelligence and Neuroscience*, 2018, 2018.
- [24] Wenyuan Xue, Qingyong Li, and Qiyuan Xue. Text detection and recognition for images of medical laboratory reports with a deep learning approach. *IEEE Access*, 8:407–416, 2019.

- [25] Jianlong Zhou, Amir H Gandomi, Fang Chen, and Andreas Holzinger. Evaluating the quality of machine learning explanations: A survey on methods and metrics. *Electronics*, 10(5):593, 2021.