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

VICTOR FUNARI TONIAL

Classificador de porte canino para apoio ao projeto Veterinários de Rua da ONG Médicos do Mundo

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

Orientador: Prof. Dr. Dennis Giovani Balreira

Porto Alegre
2023

“Não sei, só sei que foi assim!”

— CHICÓ

AGRADECIMENTOS

Gostaria de, primeiramente, agradecer aos meus pais, Roselaine Funari e Sandro Tonial, dos quais o suporte e apoio foi extremamente fundamental durante toda essa jornada.

Também é muito importante agradecer aos meus amigos e amigas, inclusive os que hoje se encontram mais distantes. A companhia de vocês fez com que esse caminho fosse muito mais agradável de se trilhar.

É importante agradecer à Camila Timm Wood e a ONG Médicos do Mundo pela motivação para o trabalho.

Por fim, não posso deixar de agradecer fortemente ao meu orientador, Dennis Giovani Balreira, que foi muito paciente e motivador durante todo o processo deste trabalho.

RESUMO

A população de animais em situação de vulnerabilidade mais do que dobrou no Brasil nos últimos anos, tornando os esforços de apoio e acolhimento a esse grupo ainda mais importantes. Com este trabalho, criamos um classificador para reconhecer o porte de caninos, a fim de auxiliar no processo de cadastro dos cães no projeto Veterinários de Rua da ONG Médicos do Mundo. Com essa classificação, os voluntários podem estimar quanto de alimento, medicamento e vacina será preciso adquirir, armazenar e levar para as ações da ONG, uma vez que não apenas o número de animais de dada região, mas também o porte de cada cão influencia nas quantidades necessárias. Para atingir esse objetivo, construímos uma rede neural profunda baseada na Inception V3, que conseguiu atingir uma acurácia de 70,6% e um F1-score de 60,1% no conjunto de dados disponibilizado pela própria ONG. Esses resultados sugerem a eficácia do classificador proposto como uma ferramenta útil na identificação dos portes dos caninos, contribuindo para o projeto.

Palavras-chave: Inteligência Artificial. CNN. Porte de Caninos. Classificação de portes.

Dog size classification to aid the Veterinários de Rua project by the World Doctors NGO

ABSTRACT

The population of vulnerable animals has more than doubled in recent years, making efforts to support this group even more important. With this work, we created a classifier to recognize the size of canines in order to assist the volunteers in the process of registering these animals in the Veterinários de Rua project, organized by the World Doctors NGO. With this classification, volunteers can estimate how much food, medicine and vaccine they will need to acquire, store and take to the NGO's actions, since not only the number of animals in a given region, but also the size of each dog influences the required quantities. To achieve this goal, we built a deep neural network based on Inception V3 that managed to reach an accuracy of 70.6% and an F1-score of 60.1% in the dataset provided by the NGO itself. These results suggest the effectiveness of the proposed classifier as a useful tool in identifying the sizes of canines, contributing to the project.

Keywords: Artificial Intelligence. CNN. Dog Size. Size Classification.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANN *Artificial Neural Network*

CNN *Convolutional Neural Network*

ONG Organização Não Governamental

SUMÁRIO

1 INTRODUÇÃO

Lidar com a população de animais em situação de vulnerabilidade é um tema muito importante na nossa sociedade. Segundo um levantamento feito em 2022 pelo Instituto Pet Brasil (IPB), o número de animais oficialmente classificados como vulneráveis mais que dobrou no país entre os anos de 2018 e 2020, indo de 3,9 milhões para 8,8 milhões (IPB, 2022). Esses números levam em conta apenas os animais que vivem sob tutela de famílias classificadas como abaixo da linha de pobreza.

1.1 Motivação

Para dar suporte a esses animais, os poderes públicos, tanto federais quanto estaduais e municipais, frequentemente apresentam programas e políticas de apoio. Por exemplo, o programa "Melhores Amigos – Bicho Sente como Gente"(Governo do Rio Grande do Sul, 2022) foi lançado em 2021 no Rio Grande do Sul com duas ações simultâneas, uma na forma de um projeto para castração de cães e gatos e outra como uma campanha contra os maus-tratos.

Ainda assim, o poder público não é a única força a dedicar recursos para essa causa. Organizações não governamentais (ONGs) também atuam e têm projetos nessa área. Dentre elas podemos citar a Médicos do Mundo (Médicos do Mundo, 2023), que, apesar de originalmente voltada apenas ao atendimento médico de seres humanos, hoje conta com um projeto intitulado Veterinários de Rua, visando suprir a demanda de atendimento médico de animais em situação de rua.

Classificar o porte dos caninos é importante tanto para a ciência quanto para a execução de políticas públicas. Estudos indicam que cães de menor porte, por exemplo, têm uma longevidade maior em relação a cães de maior porte (DEEB; WOLF, 1994). Já um estudo conduzido na cidade de São Paulo encontrou resultados divergentes (BENTUBO et al., 2007). Nele, além dos cães terem uma média de tempo de vida muito menor do que em outras cidades do mundo, os de maior porte sobrevivem por mais tempo.

Além disso, a quantidade de alimento necessário de cada cão por refeição também difere fortemente entre os múltiplos portes. Via de regra, caninos de maior porte necessitam de maior quantidade de comida durante o dia.

1.2 Objetivos

Este trabalho propõe um classificador para catalogar caninos com base em seu porte utilizando técnicas de aprendizado de máquina, a fim de auxiliar os voluntários do projeto Veterinários de Rua provendo mais agilidade no processo de cadastro dos animais. Diminuindo, assim, a carga mental exigida no primeiro contato com os tutores.

1.3 Organização

No próximo capítulo, abordaremos os conceitos básicos utilizados nesse trabalho. Começamos falando sobre a parte biológica, seguida de explicações sobre o funcionamento de redes neurais, como avaliá-las e trazendo exemplos de implementações propostas.

No Capítulo 3, é apresentada uma revisão da literatura, mostrando trabalhos que estão relacionados ao uso de aprendizado de máquina no reconhecimento de animais. Além disso, vamos falar um pouco sobre os conjuntos de dados disponíveis.

A forma com que o trabalho será realizado é explicada no Capítulo 4, onde mostramos os passos realizados para escolhermos a arquitetura da rede, pré-processamento das imagens e geração do classificador utilizado.

O Capítulo 5 apresenta os resultados obtidos, levantando discussões e análises sobre os dados finais da performance do nosso modelo. Por fim, o Capítulo 6 aborda as limitações encontradas ao executarmos este trabalho, bem como as melhorias possíveis a serem investigadas em trabalhos futuros.

2 CONCEITOS BÁSICOS

Este capítulo traz os conceitos e os fundamentos utilizados para o desenvolvimento deste trabalho. A seção que versa sobre redes neurais utiliza como base os conhecimentos do livro Inteligência artificial: uma abordagem de aprendizado de máquina (FACELI et al., 2021).

2.1 Porte dos animais

Porte é uma característica que pode ser utilizada para classificar animais no geral, como caninos, felinos, bovinos, equinos, entre outros. Não existe um consenso exato sobre quais os limites entre os portes dos caninos. Alguns estudos (BENTUBO et al., 2007) utilizam a divisão em porte pequeno como inferior a 9kg, porte médio entre 9kg e 23kg, porte grande entre 23kg e 40kg e, por fim, porte gigante acima de 40kg.

Outros estudos (MULLER; SCHOSSLER; PINHEIRO, 2008) classificam caninos como porte pequeno quando pesam até 10kg, porte médio entre 10kg e 25kg, e grandes como maiores que 25kg. Ou, até mesmo, pequeno até 15kg, médio entre 15kg e 25kg e grande sendo superior a 25kg (VENDRAMINI et al., 2022).

O projeto Veterinários de Rua da ONG Médicos do Mundo, motivadores deste trabalho, utilizam os intervalos presentes na Tabela 2.1 na classifcação dos portes.

Tabela 2.1 – Faixa de peso por porte.

Porte	Peso
PP	< 5kg
P	5kg - 15kg
M	15kg - 25kg
G	25kg - 40kg
GG	> 40kg

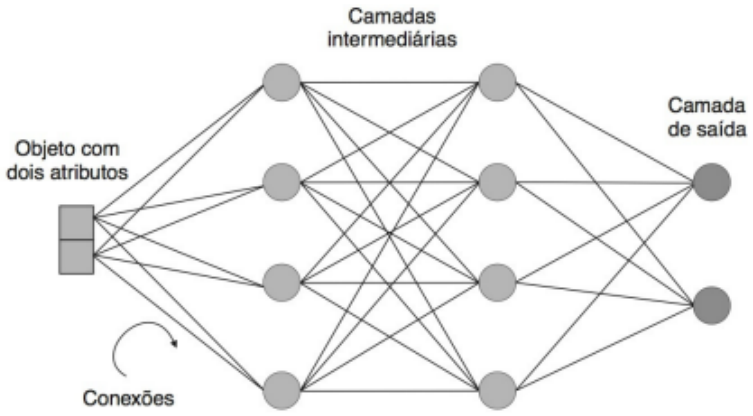
Fonte: O Autor.

2.2 Redes neurais artificiais

Redes neurais artificiais (*artificial neural network* - ANN) são modelos computacionais inspirados no funcionamento do cérebro humano. As ANNs são compostas por um conjunto interconectado de unidades de processamento chamadas de neurônios artifi-

ciais, que são organizados em camadas e trabalham em conjunto para processar informações, aprender padrões e tomar decisões. A primeira camada é conhecida como camada de entrada, a última é a camada de saída e todas as camadas entre estas duas são chamadas de camadas intermediárias (também conhecidas como camadas ocultas) (FACELI et al., 2021), como pode ser visto na Figura 2.1.

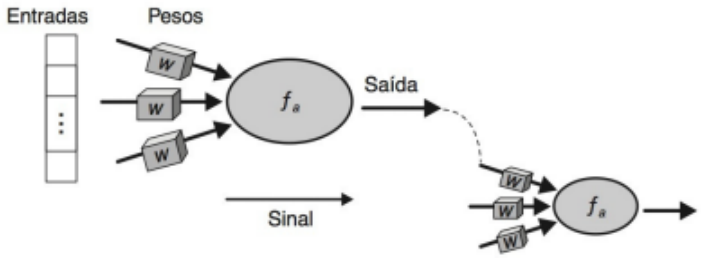
Figura 2.1 – Esquema ilustrando o funcionamento de uma rede neural artificial contendo os atributos de entrada, camadas intermediárias e camada de saída, bem como suas conexões.



Fonte: (FACELI et al., 2021).

Cada neurônio (Figura 2.2) em uma camada recebe entradas ponderadas, aplica uma função de ativação e produz uma saída que é transmitida para a próxima camada. A aprendizagem ocorre por meio do ajuste dos pesos das conexões entre os neurônios com base nos exemplos de treinamento.

Figura 2.2 – Esquema mostrando dois neurônios artificiais, evidenciando os pesos nas entradas.



Fonte: (FACELI et al., 2021).

2.2.1 Rede neural convolucional

Rede neural convolucional (*convolutional neural network* - CNN) é um tipo de arquitetura de rede neural muito utilizado para processamento de imagens e tarefas de

visão computacional. Nos últimos anos, as CNNs se tornaram o método mais utilizado para análise de imagens (SALVI et al., 2021) devido ao uso de camadas convolutivas.

Estas camadas convolutivas utilizam filtros, também chamados de *kernel*, para extrair automaticamente um mapa de características (GOODFELLOW; BENGIO; COURVILLE, 2016) específicas da imagem, como bordas e formatos. No nosso contexto de reconhecimento de caninos, uma camada pode aprender a reconhecer orelhas e focinhos enquanto outra aprende a reconhecer as texturas dos pelos.

Além das convolutivas, as CNNs também contam com camadas de *pooling*, responsáveis por reduzir a dimensionalidade das características extraídas, mantendo informações importantes. As últimas camadas destas redes, quando não são *Fully Convolutional Networks* (FCN), são camadas totalmente conectadas, onde todos os neurônios artificiais de uma camada estão conectados aos da outra. Dessa forma, todos os valores influenciam na classificação da imagem. (HUSSAIN; BIRD; FARIA, 2019)

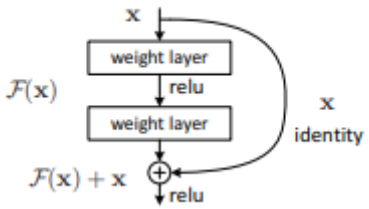
2.2.2 Arquiteturas

Existem diferentes arquiteturas de ANNs propostas, cada uma delas com um foco um pouco diferente. Dentre elas podemos citar:

ResNet: Proposta por He et al. em 2016 (HE et al., 2016), sua finalidade era tentar resolver o problema da degradação, que acontece quando redes muito profundas acabam tendo dificuldade para otimizar os pesos de todas as camadas do modelo. Dessa forma, foi proposto o uso de aprendizado residual, que consiste em agrupar camadas da rede e utilizar como saída a soma do valor da entrada entrada com o valor calculado pelo bloco.

Na Figura 2.3 podemos ver um exemplo de bloco residual, evidenciando a conexão de atalho (*shortcut connections*) utlizado para levar o valor de entrada direto para a saída do bloco.

Figura 2.3 – Figura representando um bloco residual contendo duas camadas, evidenciando a conexão de atalho.

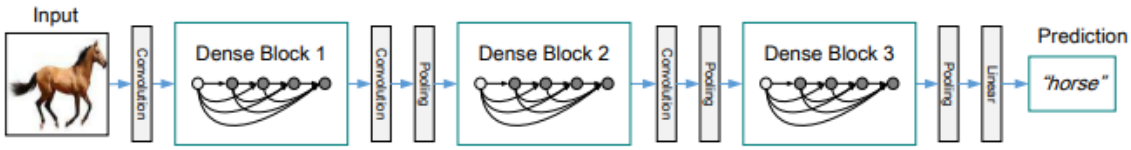


Fonte: (HE et al., 2016).

VGG: A VGG (Visual Geometry Group) (SIMONYAN; ZISSERMAN, 2014) foi proposta em 2014 como uma forma de melhorar a performance das redes convolutivas aumentando a profundidade da rede. Com 138.4 milhões de parâmetros, ela é, de longe, a arquitetura com maior número de parâmetros abordada neste trabalho.

DenseNet: Huang et al. (HUANG et al., 2017), em 2017, propuseram a DenseNet, uma rede que tenta resolver o problema da degradação a partir da criação de blocos de camadas densamente conectadas, de forma que cada camada destes blocos receba, como entrada, todas as saídas de todas as camadas anteriores, como pode ser visto na Figura 2.4.

Figura 2.4 – Diagrama mostrando a estrutura da DenseNet, evidenciando os blocos densos.



Fonte: (HUANG et al., 2017).

Inception V3: A Inception V3 (SZEGEDY et al., 2016), proposta em 2016, busca utilizar os recursos computacionais de forma mais eficiente. Para isso, fez o uso de *factorized convolutions*. Esta técnica permite a substituição de camadas convolutivas mais custosas, como com filtros 5×5 ou 7×7, por múltiplas camadas menos custosas, como as que utilizam filtros 3×3, substituindo, assim, uma única camada por uma minirrede.

Esse mesmo conceito pode ser extrapolado para que a redução seja feita para convoluções assimétricas, ou seja, transformando uma convolução de $n \times n$ em uma convolução de $1 \times n$ seguida de outra $n \times 1$.

Além disso, a Inception V3 também busca evitar o *overfitting*, que ocorre quando o modelo acaba ficando muito ajustado apenas ao conjunto de dados de treinamento, mas não se sai muito bem ao tentar classificar novos dados. Ou seja, o modelo acaba "decorando" as imagens de treinamento e não consegue generalizar para novas imagens. Para isso, foi adicionado *label-smoothing regularization*, uma técnica que encoraja o modelo a ser menos confiante em suas classificações.

2.2.3 Transfer learning

Aprendizado por transferência de conhecimento (*transfer learning*), é uma técnica onde se busca utilizar características aprendidas por uma rede neural, já previamente

treinada, durante o treinamento de outra rede (HUSSAIN; BIRD; FARIA, 2019). No lugar de treinar todo o modelo do zero para uma tarefa específica, são reutilizados pesos aprendidos anteriormente, economizando tempo e processamento (BORWARNGINN et al., 2021). A ideia é que muitas características aprendidas para realizar uma tarefa, como identificar bordas e texturas, podem ser relevantes para resolver outros problemas.

Nesta técnica, geralmente são adicionadas novas camadas com pesos aleatórios ao final da rede pré-treinada (YOSINSKI et al., 2014). Posteriormente, existe a opção de congelar os pesos das *n* camadas iniciais, garantindo sua estabilidade e focando apenas no treinamento das camadas finais do modelo. Essa abordagem visa aproveitar as características aprendidas pelas camadas pré-treinadas enquanto aprimora as capacidades do modelo para uma tarefa específica.

Como alternativa, pode-se optar por deixar que aconteça o *backpropagation* por toda a rede, permitindo que aconteça um processo chamado de *fine-tune* (YOSINSKI et al., 2014).

Manter os pesos da rede original congelados é particularmente interessante quando não se tem muitos dados disponíveis para treinamento, uma vez que treinar toda a rede em poucos dados pode gerar *overfitting*.

2.2.4 Metricas de avaliação

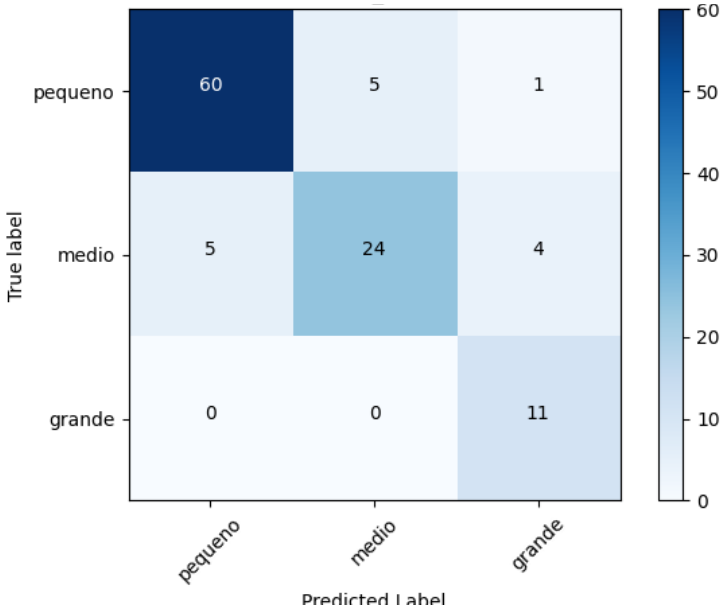
Para que se possa avaliar e comparar o desempenho de modelos de aprendizado de máquina, diferentes métricas, como acurácia, precisão, sensibilidade e *F1-score*, podem ser consideradas (THARWAT, 2020). Cada uma dessas métricas oferece uma perspectiva diferente sobre o desempenho do modelo.

Uma ferramenta poderosa para auxiliar nessa avaliação é a matriz de confusão, uma tabela onde um eixo representa a classificação real dos dados sendo avaliados e o outro a classificação prevista pelo modelo. Os números presentes nos campos da matriz são o número de ocorrências de cada combinação de classe real e prevista, como pode ser visto na Figura 2.5.

Com base na matriz de confusão, podemos extrair os seguintes valores, que são utilizados para calcularmos as métricas utilizadas na comparação entre os modelos:

- Verdadeiros Positivos (VP): Previsões corretas de que o dado analisado pertence uma dada classe.

Figura 2.5 – Exemplo de matriz de confusão multiclasse.



Fonte: O Autor.

- Falsos Positivos (FP): Previsões incorretas de que o dado analisado pertence uma dada classe.
- Verdadeiros Negativos (VN): Previsões corretas de que o dado analisado não pertence a uma dada classe.
- Falsos Negativos (FN): Previsões incorretas de que o dado analisado não pertence a uma dada classe.

Este tipo de matriz pode ser utilizada para visualizar os resultados obtidos em tarefas de classificação, podendo esse ser entre duas ou mais classes (THARWAT, 2020). Para o problema multiclasse, as métricas são calculadas individualmente para cada classe e, posteriormente, algumas abordagens podem ser tomadas para unificar essas métricas e ter uma ideia geral de como o modelo está performando. Uma dessas abordagens é o *macro-averaging* (TAKAHASHI et al., 2022), que consiste em extrair a média aritmética da métrica selecionada levando em consideração todas as classes.

Accuracy: Mais comumente utilizadas para medir a performance dos modelos (THARWAT, 2020). Ela representa a proporção entre as classificações corretas e o número total de dados. Apesar de ser muito utilizada, pode levar a análises equivocadas quando avaliada individualmente, já que é bastante sensível à *datasets* desbalanceados. A equação utilizada para calcular esta métrica é a segunte:

$$\text{Accuracy} = \frac{VP + VN}{VP + VN + FP + FN}$$

Precision: Representa a proporção entre as previsões corretas para uma dada classe em relação ao total de instâncias previstas para essa mesma classe. O cálculo dessa métrica é realizado conforme esta fórmula:

$$\text{Precision} = \frac{VP}{VP + FP}$$

Recall: Métrica que representa a proporção entre o número de previsões corretas para uma classe e o total de instâncias que realmente pertencem a ela. Sua formula é dada por:

$$\text{Recall} = \frac{VP}{VP + FN}$$

F1-score: Média harmônica entre a precisão e a sensibilidade, sendo assim, um valor mais elevado de F1-score indica que existe um equilíbrio entre estas duas métricas. A sua formula pode ser encontrada abaixo.

$$\text{F1-score} = \frac{2 * Precision * Recall}{Precision + Recall}$$

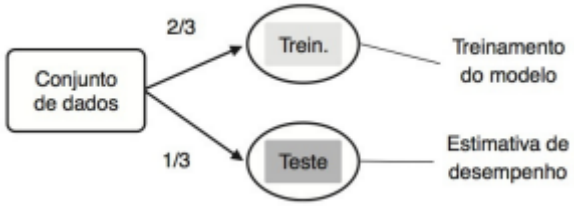
2.2.5 Métodos de amostragem

A fim de se obter essas métricas de avaliação de forma confiável, deve-se separar o conjunto de dados entre subconjuntos para treinamento e teste. Estes subconjuntos devem ser distintos para que se tenha mais confiança de que o modelo não funciona apenas para os dados utilizados no seu treinamento. Dentre as diferentes técnicas que existem para fazer essa separação, podemos citar:

***Holdout*:** Nesta técnica se divide o conjunto total de dados em dois grupos, um para treinamento, normalmente contendo dois terços dos dados, e outro para teste, como exemplificado na Figura 2.6. Uma crítica muito comum ao *holdout* é que ele não permite o entendimento de quanto uma variação nos dados de treinamento impacta no resultado final, pois é possível que o conjunto de teste fique desbalanceado, tendo apenas instâncias de uma única classe ou instâncias mais fáceis de classificar (FACELI et al., 2021).

***K-fold cross-validation*:** Uma das técnicas mais utilizadas para comparar o desempenho de modelos preditivos (RODRIGUEZ; PEREZ; LOZANO, 2009), consiste em separar o conjunto de dados em *k* grupos (*folds*), utilizando *k*-1 para treinamento e o restante para testar o modelo. No final são gerados *k* modelos, cada um treinado em um

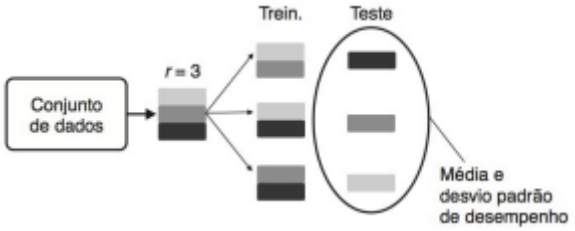
Figura 2.6 – Esquema mostrando o *holdout*, evidenciando a divisão das imagens em conjuntos de treinamento e teste.



Fonte: (FACELI et al., 2021).

subset diferente, e a performance geral é calculada como a média das k performances. Na Figura 2.7 podemos ver o conjunto de dados sendo dividido em conjuntos de treinamento e teste para um *3-fold cross-validation*.

Figura 2.7 – Divisão do conjunto de dados em um *3-fold cross-validation*.



Fonte: (FACELI et al., 2021).

2.2.6 Hiperparâmetros

Certos parâmetros não podem ser calculados diretamente a partir dos dados durante o processo de treinamento. Em vez disso, eles são definidos antecipadamente e permanecem constantes durante a execução. Eles são conhecidos como hiperparâmetros, e são responsáveis por definir a arquitetura da rede (YANG; SHAMI, 2020).

Os hiperparâmetros são importantes, também, para tentar reduzir o *overfitting*. Dentre eles, podemos citar o otimizador, o número de épocas de treinamento, a taxa de aprendizado, o número e a profundidade das camadas e o tamanho do lote (*batch*) usado para treinamento.

Otimizador: Algoritmos que buscam guiar o processo de aprendizado, determinando como o modelo atualiza os seus pesos e parâmetros a fim de reduzir o erro e aumentar a acurácia ao longo do tempo.

Existem vários algoritmos de otimização, cada um com suas próprias características e vantagens. Alguns dos otimizadores mais comuns incluem: *Stochastic Gradient Descent* (SGD) (SUTSKEVER et al., 2013), Adam (KINGMA; BA, 2014), Adafactor (SHAZEER; STERN, 2018), entre outros.

Mais especificamente, o otimizador *Adaptive Moment Estimation* (Adam) combina elementos do AdaGrad (DUCHI; HAZAN; SINGER, 2011) e do RMSProp (TIELEMAN; HINTON et al., 2012), visando adaptar a taxa de aprendizado para cada parâmetro individual do modelo, se fazendo valer do *momentum* (SUTSKEVER et al., 2013), técnica que utiliza os últimos valores de atualização do gradiente para corrigir flutuações no processo de otimização.

Épocas de treinamento: Se referem ao número de vezes que o algoritmo percorre todo o conjunto de treinamento. Um número muito grande de épocas de treinamento, além de fazer com que a execução demore mais, pode levar o modelo ao *overfitting*. Por outro lado, ter um número muito pequeno de épocas faz com que não se aprenda tanto quanto poderia ser aprendido (*underfitting*), gerando um resultado sub-ótimo.

Taxa de aprendizagem: Determina a magnitude dos ajustes que os parâmetros do modelo recebem a cada iteração. Uma taxa de aprendizado alta pode resultar em oscilações e instabilidades, fazendo com que os pesos variem muito de uma iteração para a outra. Já uma taxa muito baixa pode fazer com que o processo de treinamento fique muito lento, uma vez que os pesos são muito pouco modificados entre as iterações.

2.3 Modelos múltiplos preditivos

Modelos múltiplos preditivos (também chamado de *ensemble*) é o nome dado para modelos onde o resultado final é obtido pela combinação da decisão individual de diferentes preditores (FACELI et al., 2021). Este tipo de técnica pode ser empregada tanto para problemas de classificação quanto problemas de regressão.

É importante para um *ensemble* que os preditores que o compõem não sejam todos idênticos, de forma que cada modelo possa se especializar em identificar características específicas diferentes, ou seja, exista uma heterogeneidade nos erros de cada preditor. Desta forma, a combinação das classificações destes preditores consegue compensar eventuais erros individuais cometidos.

Para gerar a previsão final, cada uma das decisões dos preditores precisa ser levada em consideração. Das formas de combinar estes resultados, a mais comum é a votação.

Votações podem ser uniformes, onde todos os votos contribuem igualmente para a classificação final, ou com pesos, onde cada classificador tem um peso associado, fazendo com que o voto de uns valha mais que o dos outros.

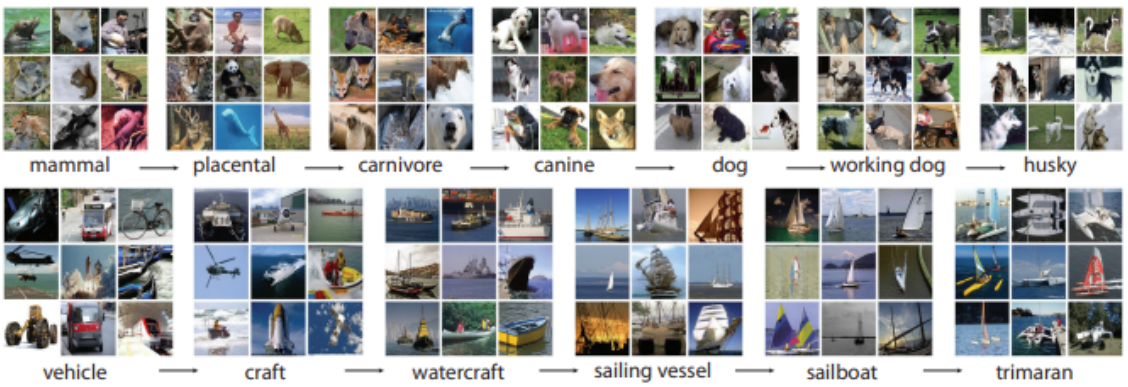
Uma das técnicas que pode ser utilizada para criar este tipo de modelo é o *Bootstrap aggregating* (ou *Bagging*), que utiliza a combinação de classificadores homogêneos, ou seja, combinação de classificadores gerados por um único algoritmo. Nesta técnica são produzidas réplicas do conjunto de treinamento utilizando amostragem com reposição e, para cada uma destas réplicas, um classificador diferente é treinado. O resultado final deste tipo de modelo é obtido, mais comumente, utilizando voto uniforme (FACELI et al., 2021), onde todos tem o mesmo peso.

3 TRABALHOS RELACIONADOS

A classificação de imagens utilizando aprendizado de máquina é um tópico muito importante e pesquisado, sendo relevante tanto para tarefas críticas de visão computacional, como carros autônomos e sistemas de segurança, quanto outras menos críticas, como encontrar alguma foto específica na sua galeria de fotos.

Com o intuito de dar suporte às pesquisas na área de visão computacional e reconhecimento de objetos, o trabalho de Deng et al. (DENG et al., 2009) introduziu a ImageNet, uma ontologia de imagens de larga escada, contendo 3,2 milhões de imagens divididas em 5247 categorias, como pássaros, móveis, mamíferos e veículos. A Figura 3.1 mostra exemplos das imagens catalogadas nesta ontologia.

Figura 3.1 – Exemplo de imagens categorizadas pela ImageNet, agrupadas em 13 classes.



Fonte: (DENG et al., 2009).

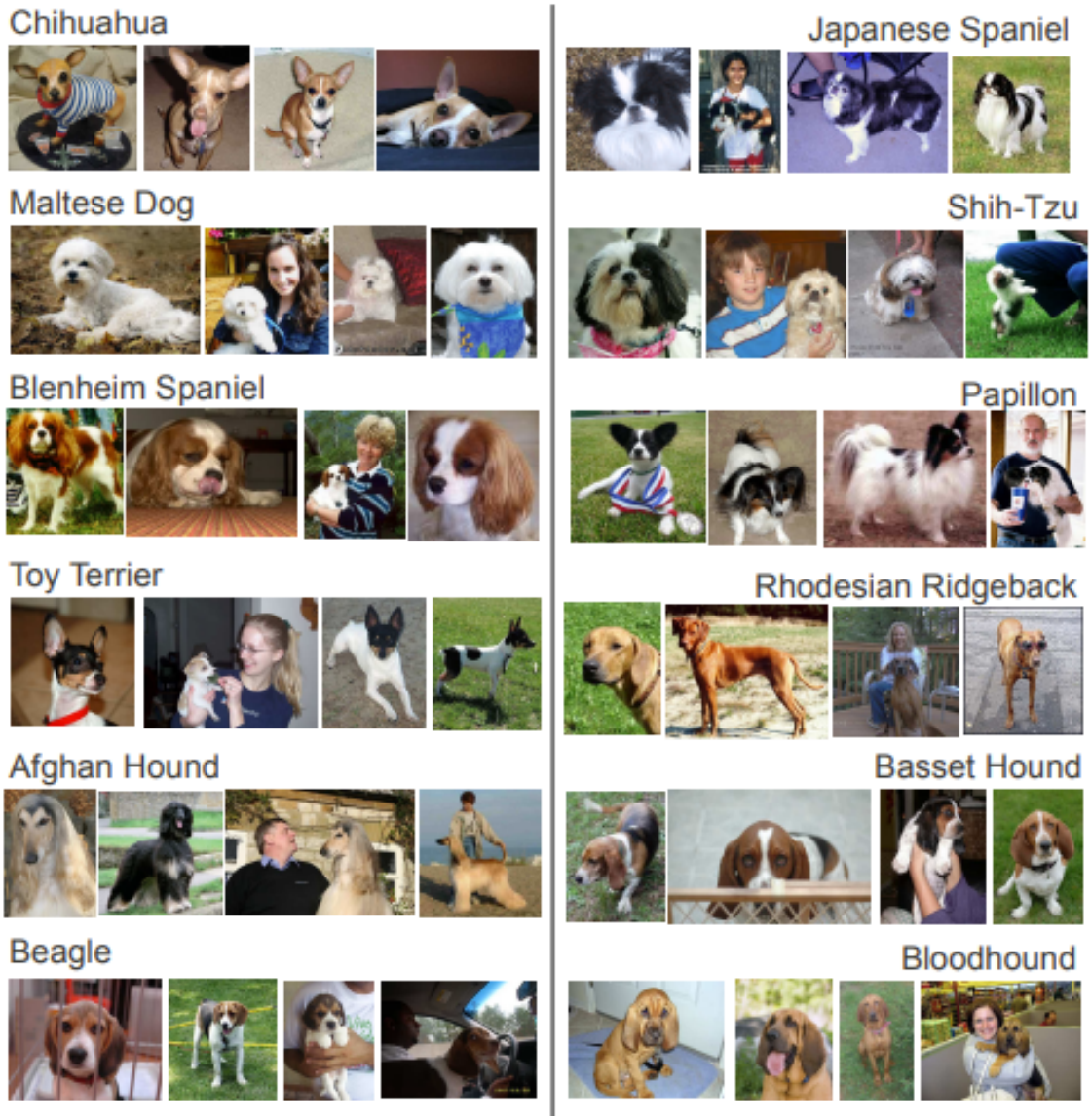
Existem, também, outros *datasets* mais especializados. Por exemplo, o trabalho de Welinder et al. (WELINDER et al., 2010) disponibilizou um *dataset* com mais de 6 mil imagens, pertencendo a 200 espécies de pássaros.

Já o trabalho de Khosla et al. (KHOSLA et al., 2011) criou o Stanford Dogs (Figura 3.2), uma base de dados contendo 22 mil imagens de caninos, divididas em 120 raças com base nas imagens presentes no ImageNet.

Dentro desta área, o reconhecimento de animais vem sendo bastante estudado. Em 2022 o trabalho de Gupta et al. (GUPTA et al., 2022) conseguiu resultados interessantes utilizando uma YOLOv4 (BOCHKOVSKIY; WANG; LIAO, 2020) para realizar o reconhecimento raças de bovinos com base em imagens.

A classificação de caninos por raças também é um ponto de interesse das pesquisas. Em 2018, o trabalho de Ráduly et al. (RÁDULY et al., 2018) comparou o uso de uma NASNet-A mobile (ZOPH et al., 2018) e uma Inception-ResNet-v2 (SZEGEDY et

Figura 3.2 – Exemplo de imagens do Stanford Dogs evidenciando 12 raças de cães.



Fonte: (KHOSLA et al., 2011).

al., 2017), ambas pré-treinadas na ImageNet (DENG et al., 2009). Para comparar os desempenhos foi utilizado *5-fold cross-validation*, atingindo, respectivamente, acurácias de 80,72% e 90,69% no Stanford Dogs (KHOSLA et al., 2011).

Em 2021, o trabalho realizado por Borwarnginn et al. (BORWARNGINN et al., 2021) propôs realizar a identificação das raças de caninos utilizando apenas a imagem das suas faces, fazendo o uso de CNNs para atingir esse objetivo. Nesse trabalho, os autores compararam as performances das arquiteturas MobilenetV2 (SANDLER et al., 2018), InceptionV3 e NASNet. No seu estudo, utilizaram a técnica de *transfer learning* para aproveitar as características aprendidas pelas redes no *dataset* ImageNet, treinando apenas a camada de classificação das redes no Columbia Dogs (LIU et al., 2012). Por

considerar que tinham poucas imagens em relação à quantidade de classes presentes no seu *dataset*, aplicaram técnicas de *data augmentation* para, artificialmente, aumentar este número.

Para avaliar os desempenhos das redes, os autores dividiram o seu conjunto de dados em dois. Um para treinamento e outro para teste. Conseguindo atingir, no conjunto de teste, acurácias de até 89,02% para a InceptionV3, 81,65% na MobilenetV2 e, por fim, 89,92% na NASNet.

Já o trabalho de Dabrowski et al. (DABROWSKI et al., 2021) usou uma rede construída tendo uma Inception V3 como base para realizar o reconhecimento de raças de caninos. Essa rede foi treinada utilizando as imagens do *Stanford Dogs*, fazendo uso de *data augmentation* por terem julgado 20.580 imagens um número muito pequeno. Os autores optaram por realizar 10 épocas de treinamento no modelo e utilizaram o otimizador Adam com taxa de aprendizagem de 0,0001. Com estes parâmetros, conseguiram inicialmente uma acurácia de 67,9%, que foi melhorada para 78,0% com a adição de mais camadas na rede.

O trabalho de Madhan et al. (MADHAN; KAUSHIK; RAJU, 2022) propôs atingir uma melhor acurácia usando a menor quantidade de recursos computacionais possível para identificar raças de caninos com base em suas imagens. O primeiro modelo gerado nesse trabalho foi uma rede que utilizava o otimizador RMSprop, atingindo uma acurácia de 64%. Para tentar melhorar a performance, foi utilizada uma implementação da mobileNetV2 pré-treinada na ImageNet e, usando a técnica de *transfer learning*, os autores treinaram apenas as últimas camadas para identificar as raças de cães, atingindo uma acurácia de 77%.

Após analisarem o desempenho, os autores entenderam que a rede estava sofrendo de *overfitting* e, em busca de tentar resolver este problema, modificaram o otimizador para o Adam. Esta escolha se mostrou efetiva, fazendo com que a rede conseguisse atingir uma acurácia de 81% no conjunto de validação.

Outras abordagens também foram testadas, como realizar a biometria da face para a identificação individual dos caninos (MOUGEOT; LI; JIA, 2019), reconhecer as espécies e raças de animais com base no seu som (PABICO et al., 2015) e, até mesmo, identificar o contexto no qual o cão estava (em ações como "brincando", "brigando" e outros) utilizando o som do seu latido (MOLNÁR et al., 2008).

Ao analisarmos a literatura, até onde nos foi possível investigar, não encontramos nenhum trabalho de classificação catalogando caninos com base em seu porte, seja ele

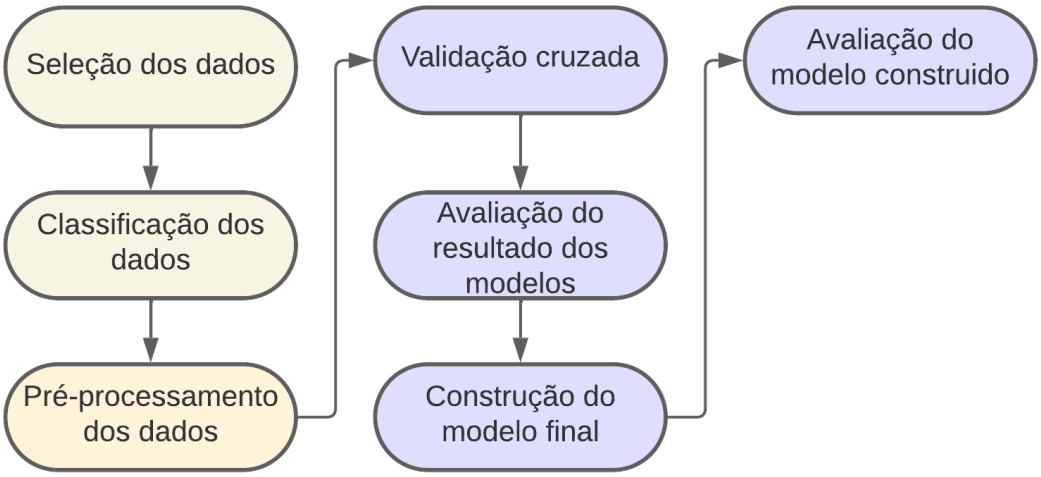
com ou sem raça definida. Todos os trabalhos disponíveis versavam ou sobre classificar a raça do animal, ou realizar a identificação individual do mesmo. É interessante notar também que os trabalhos avaliados neste capítulo utilizaram sempre a acurácia como medida de desempenho. Por sabermos que a acurácia é uma métrica bastante sensível a conjuntos de dados desbalanceados, apresentamos também o *F1-score*.

Outro ponto recorrente nesses trabalhos é o fato de que se tem poucas imagens nos *datasets* utilizados. Dessa forma, diversos desses trabalhos utilizam técnicas de *data augmentation* para tentar lidar com esse fato. Ainda assim, é recorrente a conclusão de que *datasets* maiores teriam um impacto positivo na acurácia da rede.

4 METODOLOGIA

Neste capítulo vamos abordar qual foi a metodologia escolhida para classificar os caninos, começando pela forma de seleção do *dataset*, passando pelos testes que foram realizados a fim de definir qual o modelo mais apropriado para a implementação e, por fim, criar o nosso classificador. A ordem dos passos tomados neste trabalho pode ser visualizada na Figura 4.1.

Figura 4.1 – Fluxograma da metodologia do trabalho, dividido em duas partes. A primeira trata da obtenção e preparação dos dados, a segunda da criação e avaliação do classificador.



Fonte: O Autor.

4.1 *Dataset*

Para realizar este trabalho, contamos com um *dataset* proprietário, cedido pela ONG Médicos do Mundo, se tornando uma parceira para o desenvolvimento deste trabalho. O *dataset* em questão contém 68 imagens de caninos, sendo elas, em sua grande maioria, de caninos sem raça definida, visto que a ONG atende a população de rua da cidade de Porto Alegre. É importante salientar que as fotos disponíveis no conjunto da ONG não possuíam nenhum padrão, isto é, existiam imagens de animais de frente, de lado, de costas e até no colo de pessoas, além de variarem também as distâncias do cão para a câmera e resoluções das imagens.

Vale ressaltar, também, que o *dataset* providenciado pela ONG é desbalanceado. A maioria dos animais atendidos são de porte pequeno, tendo 45 imagens para esse rótulo,

seguido de porte médio com 21 fotos. Por fim, foi disponibilizado apenas um exemplar para os portes grande e gigante, cada.

Para que fosse possível uma melhor classificação, escolhemos enriquecer o *dataset* da ONG com mais imagens, a fim de que a rede neural tivesse dados mais balanceados e em maior quantidade para utilizar no período de treinamento.

Dada a falta de imagens de caninos pré-classificadas por porte disponível na internet, tivemos de classificar manualmente, foto a foto, parte do Stanford Dogs (KHOSLA et al., 2011), um subset do ImageNet (DENG et al., 2009). Esta classificação foi revisada por uma bióloga voluntária da ONG.

Ao total foram catalogadas e adicionadas 101 imagens. Exemplos destas imagens podem ser encontradas na Figura 4.2.

Figura 4.2 – Imagens do Stanford Dogs.

Fonte: (KHOSLA et al., 2011).

Tendo em vista que o nosso *dataset* não contém muitas imagens para cada uma

das possíveis classificações de porte (PP, P, M, G e GG), optamos por condensar as classes mais extremas (PP e P para P e G e GG para G) em uma só, ficando com P, M e G.

Com isso, temos um total de 169 imagens compondo o nosso *dataset* final, sendo essas imagens divididas em 76 de porte pequeno, 47 médio e 46 grande.

Em respeito à privacidade garantida pela ONG aos tutores dos animais, ficamos legalmente proibidos de divulgar publicamente as imagens presentes no seu banco de dados.

4.2 Classificador

Inicialmente, cogitamos desenvolver e treinar uma rede neural convolucional personalizada. Entretanto, descartamos rapidamente esta abordagem devido ao fato de possuirmos um *dataset* de tamanho muito reduzido para trabalhar. Dessa forma, optamos por uma estratégia mais vantajosa e bastante utilizada na literatura, chamada *transfer learning*, que nos permite aproveitar características já aprendidas por um modelo pré-existente. Assim, concentramos nossos esforços em treinar apenas a camada de saída para classificar o porte de caninos utilizando nosso próprio conjunto de dados. Essa decisão se mostrou mais interessante, pois nos permitiu aproveitar o conhecimento prévio de *features* treinadas em uma base de dados muito maior, a ImageNet, reduzindo o esforço computacional necessário e, ao mesmo tempo, obtendo resultados muito melhores.

Há uma ampla diversidade de modelos de redes neurais pré-treinadas disponíveis para seleção. A fim de identificarmos o modelo mais adequado para realizar este trabalho, diversos testes foram realizados. Para isso, o máximo de hiperparâmetros foi mantido fixo, como o conjunto de dados utilizado, o número de *folds* para validação cruzada em 5, o algoritmo de otimização empregado como ADAM, a taxa de aprendizado adotada em 0,0001, o número de épocas de treinamento em 20 e a função de perda (*loss function*) *categorical cross entropy*. O *dataset* utlizado no treinamento e teste foi o apresentado na Seção 4.1, sem *data augmentation*. Essa abordagem teve como intuito garantir que o único elemento variável entre os diferentes testes fosse o modelo da rede neural em consideração. Esses valores foram definidos de forma empírica, baseados nos escolhidos e testados pelos trabalhos apresentados no Capítulo 3.

A título de comparação, testamos a nossa própria rede CNN personalizada (aqui referenciada como Customizada), que conta com 8 camadas ocultas e 1,5M parâmetros, comparando com classificadores criados utilizando *transfer learning* baseados nos mode-

los já implementados e treinados disponíveis no Keras (CHOLLET et al., 2015).

Os modelos utilizados para comparação incluem a ResNet50, que contém 25,6M parâmetros, a DenseNet201 com 20,2M parâmetros, a Inception V3 com 23,9M parâmetros e a VGG16, que possui 138,4M parâmetros.

Tabela 4.1 – Comparação de desempenho entre os modelos.

Modelo	Acurácia	Precisão	Sensibilidade	F1-score
Customizada	0,467	0,472	0,255	0,408
Resnet50	0,491	0,548	0,356	0,454
Densenet201	0,615	0,708	0,390	0,568
Inception V3	**0,705**	**0,771**	0,622	**0,675**
VGG16	0,674	0,683	**0,656**	0,650

Fonte: O Autor.

Como podemos ver na Tabela 4.1, a nossa implementação utilizando Inception V3 acabou produzindo tanto o F1-score quanto a acurácia melhor que as outras implementações. Optamos por dar continuidade no trabalho com esta arquitetura.

As matrizes de confusão utilizadas no cálculo destas métricas podem ser encontradas no Apêndice A.

4.3 Pré-processamento das imagens

Antes de começar o treinamento da camada de saída do nosso modelo, foram realizados ajustes nas imagens. Inicialmente, as imagens foram redimensionadas para uma resolução de 224×224 pixels, a fim de padronizar as nossas entradas. Este passo foi importante uma vez que as imagens do nosso *dataset* tinham resoluções distintas e, também, para que o processo de treinamento fosse mais rápido, já que a rede teria menos parâmetros para aprender (SAPONARA; ELHANASHI, 2021). Este valor foi originalmente escolhido com base no valor padrão de entrada das redes VGG, DenseNet e ResNet utilizadas neste trabalho.

Além disso, foi aplicada a técnica de *zero-centering* em cada canal de cor das imagens. Essa técnica subtrai a média dos valores de cada um dos canais em busca de centralizar seus valores ao redor de zero, o que contribui para um treinamento mais eficiente e melhora o desempenho do modelo. Exemplos destas imagens podem ser encontrados na Figura 4.3.

É importante indicar que, durante o processo dos testes para selecionarmos a arquitetura com a qual seguiríamos o trabalho, algumas técnicas específicas, que são requi-

Figura 4.3 – Imagens pré-processadas.

Fonte: O Autor.

sitos dos diferentes modelos, foram utilizadas. Dentre elas podemos citar:

Alteração do espectro de cores RGB para BGR: Inverte a ordem dos canais de cores, sendo útil para redes que foram treinadas previamente utilizando esta organização. No caso deste trabalho, tanto a implementação de VGG quanto a ResNet, disponíveis no Keras, necessitavam desta modificação.

Normalização dos canais de cores: Ajusta todos os valores dos canais de cores para ficarem entre zero e um, tendo um intuito parecido com o *zero-centering*, mas sem mudar a distribuição dos valores. Esta técnica é necessária no pré-processamento das imagens da DenseNet.

4.4 Geração do modelo preditivo

Por fim, tendo em vista o nosso *dataset* de tamanho reduzido, escolhemos explorar uma abordagem que conseguisse lidar melhor com modelos que sofrem de *overfitting*, como o *bootstrap aggregating*.

Para tal, optamos por manter os mesmos hiperparâmetros utilizados nos testes

sempre que possível, sendo eles otimizador ADAM com *learning rate* de 0,0001 e 20 épocas de treinamento. Além disso, por estarmos utilizando o *bootstrap aggregating*, é importante definir a quantidade de modelos que serão incluídos no nosso *ensemble*.

Começamos utilizando 10 modelos, cada um treinado ao longo de 20 épocas. Para o treinamento, selecionamos 120 amostras, retirando-as com reposição, de um conjunto de imagens que contém 80% do nosso *dataset*. Os 20% restantes não foram utilizados durante o processo de treinamento, sendo reservados exclusivamente para a validação dos resultados obtidos. Nos referenciaremos a este conjunto como *holdout*.

Em busca de resultados melhores, optamos por aumentar o número de modelos que compõem o nosso *ensemble*. Para isso, foi necessário mudar a abordagem, já que a máquina que tínhamos à disposição não conseguia comportar mais de 15 modelos em memória. Assim, passamos a realizar o treinamento e execução em *batches* de modelos, treinando 10 de cada vez e os salvando em disco quando não estavam mais sendo utilizados.

Este trabalho foi desenvolvido utilizando um sistema Windows com uma CPU Intel Core I5 13600k, 16 GB RAM DDR4 e uma GPU NVIDIA GeForce RTX 2070 SUPER, que acabou não sendo utilizada. Todo o treinamento e teste foi processado diretamente pela CPU.

Por fim, usamos a mesma lógica na hora de avaliar os resultados. Em cada iteração, 10 modelos votaram por vez, até que todos os modelos criados e armazenados tivessem dado o seu veredito, então todos os votos foram somados e a classificação da maioria era escolhida.

É importante ressaltar que, apesar de ter nos propiciado esse expressivo aumento no número de modelos, utilizar o disco tornou tanto o processo de execução quanto o de treinamento consideravelmente mais lento.

5 RESULTADOS E DISCUSSÃO

Para termos uma boa ideia da evolução e conseguirmos visualizar na prática o efeito da sabedoria das multidões (SUROWIECKI, 2005) do nosso *ensemble*, treinamos, em um primeiro momento, o classificador com apenas um modelo. Os resultados deste teste podem ser encontrados na Tabela 5.1.

A expectativa, que acabou se concretizando, é que o desempenho não fosse satisfatório. Esperávamos este resultado por conta do nosso conjunto de dados total ser muito pequeno, levando mais facilmente o modelo ao *overfitting*.

Optamos por utilizar tanto o *F1-score* quanto a acurácia como métricas para comparar os classificadores implementados, escolhemos estas duas métricas por entender que apenas a acurácia não seria suficiente, dada a natureza desbalanceada do nosso *dataset*. Como o nosso problema é multiclasse, utilizaremos o macro *F1-score* nas comparações.

Testamos os nossos classificadores contra algumas combinações diferentes de conjunto de dados, sendo elas:

Somente as imagens do *Stanford Dogs*: Fotos de cães com raça definida.

Somente imagens da ONG: Fotos de cães sem raça definida e com bastante ruído.

Combinação das imagens de Stanford e da ONG: Combinação das imagens dos *datasets* de Stanford e da ONG, do qual 80% foi utilizado para o treinamento.

Conjunto de holdout: Combinação das imagens dos *datasets* do *Stanford Dogs* e da ONG que não foram usadas no treinamento.

Escolhemos estas divisões para testar a hipótese de que, por conta da rede Inception V3 utilizada no *transfer learning* ter sido pré-treinanda utilizando as imagens contidas no ImageNet, o desempenho do nosso classificador seria melhor ao avaliar o conjunto de Stanford individualmente, visto que este conjunto é um subconjunto da ImageNet.

As performances mais relevantes para este trabalho são os resultados do *dataset* de *holdout*, por conter apenas imanges não utilizadas nos testes, e do *dataset* próprio da ONG, por ser o objetivo final do classificador.

Ao analisarmos os resultados para as implementações de 1 modelo (Tabela 5.1), 10 modelos (Tabela 5.2) e 100 modelos (Tabela 5.3), podemos verificar que o aumento no número de modelos presentes no *ensemble* melhora o macro *F1-score* do classificador de 40% para 90,2% no conjunto de *holdout*. Esse comportamento é justamente o que esperávamos, uma vez que, com mais modelos, o viés e os erros individuais passam a ser menos importantes.

Tabela 5.1 – Desempenho Inception V3 com 1 modelo e 20 épocas.

Dataset avaliado	Acurácia	F1-score
Stanford Dogs	0,636	0,531
ONG	0,632	0,414
Stanford Dogs + ONG	0,639	0,626
Holdout	0,411	0,400

Fonte: O Autor.

Tabela 5.2 – Desempenho Inception V3 com 10 modelos e 20 épocas.

Dataset avaliado	Acurácia	F1-score
Stanford Dogs	0,909	0,882
ONG	0,720	0,576
Stanford Dogs + ONG	0,864	0,856
Holdout	0,735	0,724

Fonte: O Autor.

Tabela 5.3 – Desempenho Inception V3 com 100 modelos e 20 épocas.

Dataset avaliado	Acurácia	F1-score
Stanford Dogs	0,867	0,835
ONG	0,706	0,601
Stanford Dogs + ONG	0,876	0,869
Holdout	0,912	0,902

Fonte: O Autor.

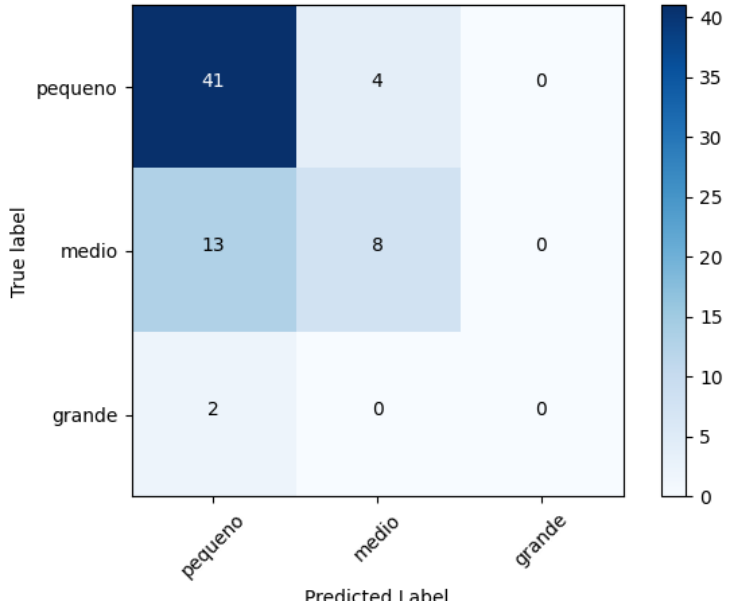
Outro dado interessante de ser analisado é a evolução do *F1-score*, de 57,6% para 60,1% obtido no conjunto de dados da ONG, ao passarmos de 10 modelos para 100, com uma piora de 2% na acurácia.

Analisando as matrizes de confusão deste caso (presentes na Figura 5.1 e Figura 5.2), podemos notar que isso se deu pois, apesar de ter classificado corretamente uma imagem a mais, o classificador com 10 modelos não acertou nenhuma previsão para a classe de porte grande.

Assim, resolvemos investir mais em variações do cenário de melhor performance, presente na Tabela 5.3. Com o intuito de verificar se não estávamos sofrendo de *underfitting* nos nossos classificadores, optamos por aumentar o número de épocas de treinamento de cada modelo de 20 para 30 (Tabela 5.4) e 40 (Tabela 5.5), mantendo fixa a quantidade de integrantes e demais hiperparâmetros do *ensemble*.

Comparando os desempenhos dos melhores classificadores, os que utilizam 100 modelos no *ensemble*, conseguimos notar que a utilização de mais épocas de treinamento melhora os resultados nos *datasets* completos (dos quais 80% foi usado no treinamento), mas piora o resultado no conjunto de *holdout*, que garantidamente não participou do pro-

Figura 5.1 – Matriz de confusão 10 modelos e 20 épocas no *dataset* da ONG.



Fonte: O Autor.

Tabela 5.4 – Desempenho Inception V3 com 100 modelos e 30 épocas.

Dataset avaliado	Acurácia	F1-score
Stanford Dogs	0,873	0,837
ONG	0,750	0,628
Stanford Dogs + ONG	0,893	0,889
Holdout	0,795	0,794

Fonte: O Autor.

cesso de treinamento dos modelos.

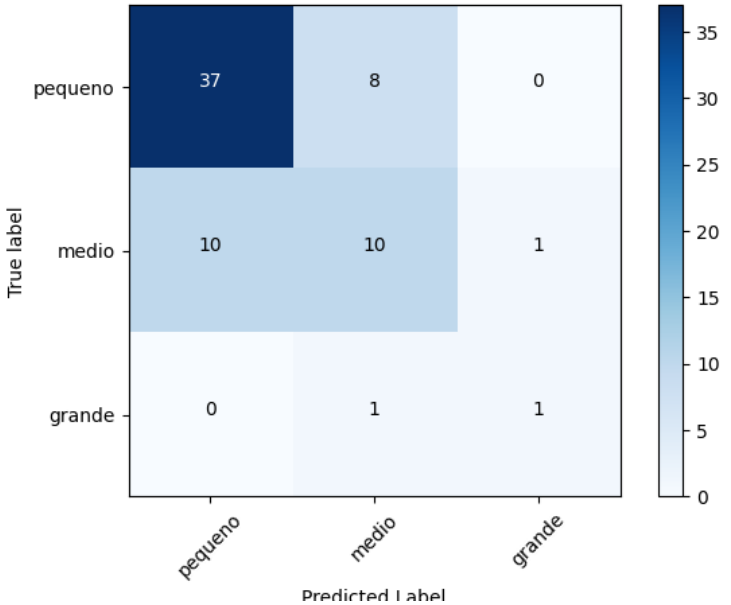
Esse comportamento pode ser causado por dois motivos distintos: podemos estar, ao aumentar o número de épocas de treinamento, levando todos os modelos muito ao *overfitting*, ou, então, podemos ter tido o infortúnio do conjunto de *holdout* do classificador treinado usando apenas 20 épocas ser muito mais fácil que os outros.

Entendemos, após identificar um padrão de diminuição da performance no conjunto de *holdout* com o aumento do número de épocas, que a primeira hipótese é mais provável. Entretanto, são necessários mais testes para ser afirmar categoricamente que este é o caso.

Apesar de não ter uma performance de classificador estado da arte, consideramos os resultados obtidos pelo nosso classificador, que utiliza 100 modelos treinados por 20 épocas, muito satisfatórios. O macro *F1-score* de 90,2% no conjunto de *holdout* nos dá confiança de que os resultados obtidos não são aleatórios. A matriz de confusão deste classificador, quando avaliando o conjunto de *holdout*, pode ser encontrada na Figura 5.3.

Ainda assim, existe muito espaço para melhoria. A adição de mais dados ao *da-*

Figura 5.2 – Matriz de confusão 100 modelos e 20 épocas no *dataset* da ONG.



Fonte: O Autor.

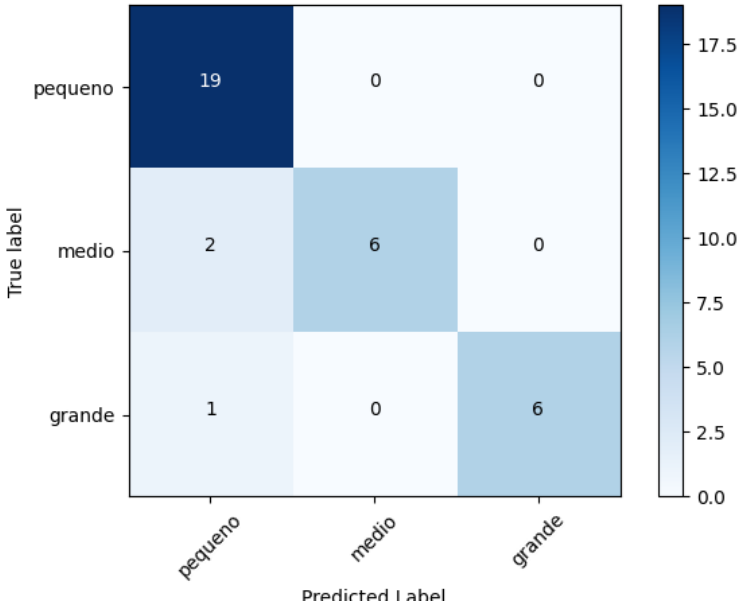
Tabela 5.5 – Desempenho Inception V3 com 100 modelos e 40 épocas.

Dataset avaliado	Acurácia	F1-score
Stanford Dogs	0,890	0,863
ONG	0,750	0,649
Stanford Dogs + ONG	0,905	0,902
Holdout	0,706	0,699

Fonte: O Autor.

taset da ONG, com imagens de caninos sem raça definida, pode auxiliar tanto no treinamento da rede quanto na própria avaliação dos resultados. Acreditamos que, com mais imagens nos conjuntos de teste, teríamos uma confiança ainda maior no valor das métricas obtidas.

Figura 5.3 – Matriz de confusão 100 modelos e 20 épocas no *dataset* de *holdout*.



Fonte: O Autor.

6 CONCLUSÃO

Este trabalho se dedicou a gerar um classificador, utilizando aprendizado de máquina, para identificar o porte de caninos com base em sua imagem. Para isso, além de ter acesso à parte do banco de dados da ONG Médicos do Mundo, classificamos manualmente parte do *Stanford Dogs Dataset*. De posse destes dados, investigamos algumas arquiteturas já propostas em busca do melhor candidato para o nosso classificador.

Por fim, foi criado um modelo *ensemble* com 100 participantes, atingindo resultados interessantes tanto para o conjunto de teste (nomeado como *holdout* no capítulo 5) quanto para o conjunto de dados providenciado pela ONG parceira. Atingindo um *F1-score* de 90,2% e acurácia de 91,2% no conjunto de *holdout* e, respectivamente, 60,1% e 70,6% no conjunto de dados da ONG.

Consideramos estes resultados bastante positivos, visto que o tamanho total do nosso *dataset* utilizado para treinar a rede era bem diminuto.

6.1 Limitações

O trabalho sofreu influência da pouca quantidade de dados disponível em pelo menos duas frentes.

A primeira é que os resultados podem melhorar a medida que mais dados forem adicionados ao treinamento da rede, inclusive fazendo com que os hiperparâmetros possam ser alterados para que a rede, por exemplo, treine por mais épocas ou que sejam adicionadas mais camadas ao final da Inception V3 utilizada.

A segunda é que, por termos um conjunto tão pequeno de imagens, as métricas obtidas ao final do trabalho podem ter sido afetadas. Acreditamos que, ao adicionar mais dados ao nosso *dataset*, é possível uma maior confiança nos resultados.

Outro fator que pode afetar a confiança nos resultados obtidos no conjunto de *holdout* é o de que a implementação de Inception V3 utilizada no *transfer learning* foi pré-treinada na ImageNet, *dataset* do qual o *Stanford Dogs* é um *subset*.

Além disso, no âmbito dos dados utilizados, o conjunto de fotos que continha os cães sem raça definida (disponibilizados pela ONG) não possuía nenhum padrão. As imagens continham caninos nas mais variadas posições e distâncias da câmera, incluindo no colo dos tutores, de costas para a câmera, deitados, de pé, com e sem outros elementos na foto (como mesas e cadeiras).

Outra limitação encontrada foi no poder computacional disponível para o treinamento e execução dos modelos. O salto de utilizar um *ensemble* de 10 modelos para um de 100 foi bem significativo em questão de tempo, já que agora o disco estava sendo usado durante o processo. Esse fato acabou sendo um impeditivo para que fossem realizados novos testes aumentando a quantidade de modelos.

6.2 Trabalhos futuros

Para dar seguimento a este trabalho, algumas melhorias podem ser realizadas. Seria importante trabalhar na obtenção de mais dados de caninos sem raça definida, classificados por porte, para usar tanto no treinamento quanto no teste da rede.

Concomitantemente, pode ser alinhado um padrão a ser utilizado nas fotos, com o objetivo de que a rede tenha menos trabalho para, inclusive, identificar caninos nas imagens.

Ainda assim, outra abordagem que poderia ajudar a mitigar este problema seria a utilização de *data augmentation*, aumentando, assim, o número de imagens disponíveis de maneira artificial. Estas abordagens podem utilizar rotações e alteração de cores nas imagens originais, por exemplo.

REFERÊNCIAS

BENTUBO, H. D. L. et al. Expectativa de vida e causas de morte em cães na área metropolitana de são paulo (brasil). **Ciência Rural**, SciELO Brasil, v. 37, p. 1021–1026, 2007.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. Yolov4: Optimal speed and accuracy of object detection. **arXiv preprint arXiv:2004.10934**, 2020.

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.

DABROWSKI, A. et al. Dog breed library with picture-based search using neural networks. In: IEEE. **2021 IEEE 16th International Conference on Computer Sciences and Information Technologies (CSIT)**. [S.l.], 2021. v. 1, p. 17–20.

DEEB, B.; WOLF, N. Studying longevity and morbidity in giant and small breeds of dogs. **Vet Med**, v. 89, n. suppl, p. 702–713, 1994.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. **2009 IEEE conference on computer vision and pattern recognition**. [S.l.], 2009. p. 248–255.

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.

FACELI, K. et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. [S.l.]: LTC, 2021.

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

Governo do Rio Grande do Sul. **Programa inédito reforçará políticas públicas de proteção animal no RS**. 2022. Acessado em: 19 de agosto de 2023. Available from Internet: <https://estado.rs.gov.br/programa-inedito-reforcara-politicas-publicas-de-protecao-animal-no-rs>.

GUPTA, H. et al. Computer vision-based approach for automatic detection of dairy cow breed. **Electronics**, MDPI, v. 11, n. 22, p. 3791, 2022.

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

HUANG, G. et al. Densely connected convolutional networks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2017. p. 4700–4708.

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.

IPB. **Número de animais de estimação em situação de vulnerabilidade mais do que dobra em dois anos, aponta pesquisa do IPB**. 2022. Acessado em: 19 de agosto de 2023. Available from Internet: <http://institutopetbrasil.com/fique-por-dentro/numero-de-animais-de-estimacao-em-situacao-de-vulnerabilidade-mais-do-que-dobra-em-dois-anos-aponta-pesquisa-do-ipb/>.

KHOSLA, A. et al. Novel dataset for fine-grained image categorization: Stanford dogs. In: CITESEER. **Proc. CVPR workshop on fine-grained visual categorization (FGVC)**. [S.l.], 2011. v. 2, n. 1.

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

LIU, J. et al. Dog breed classification using part localization. In: SPRINGER. **Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12**. [S.l.], 2012. p. 172–185.

MADHAN, E.; KAUSHIK, A.; RAJU, R. An intelligent dog breed recognition system using deep learning. **International Journal of Data Informatics and Intelligent Computing**, v. 1, n. 1, p. 39–51, 2022.

MOLNÁR, C. et al. Classification of dog barks: a machine learning approach. **Animal Cognition**, Springer, v. 11, p. 389–400, 2008.

MOUGEOT, G.; LI, D.; JIA, S. A deep learning approach for dog face verification and recognition. In: SPRINGER. **PRICAI 2019: Trends in Artificial Intelligence: 16th Pacific Rim International Conference on Artificial Intelligence, Cuvu, Yanuca Island, Fiji, August 26-30, 2019, Proceedings, Part III 16**. [S.l.], 2019. p. 418–430.

MULLER, D. C. d. M.; SCHOSSLER, J. E.; PINHEIRO, M. Adaptação do índice de massa corporal humano para cães. **Ciência Rural**, SciELO Brasil, v. 38, p. 1038–1043, 2008.

Médicos do Mundo. **sonre nós**. 2023. Acessado em: 19 de agosto de 2023. Available from Internet: <https://www.medicosdomundo.org.br/parceiros-do-bem/about/>.

PABICO, J. P. et al. Automatic identification of animal breeds and species using bioacoustics and artificial neural networks. **arXiv preprint arXiv:1507.05546**, 2015.

RÁDULY, Z. et al. Dog breed identification using deep learning. In: IEEE. **2018 IEEE 16th International Symposium on Intelligent Systems and Informatics (SISY)**. [S.l.], 2018. p. 000271–000276.

RODRIGUEZ, J. D.; PEREZ, A.; LOZANO, J. A. Sensitivity analysis of k-fold cross validation in prediction error estimation. **IEEE transactions on pattern analysis and machine intelligence**, IEEE, v. 32, n. 3, p. 569–575, 2009.

SALVI, M. et al. The impact of pre-and post-image processing techniques on deep learning frameworks: A comprehensive review for digital pathology image analysis. **Computers in Biology and Medicine**, Elsevier, v. 128, p. 104129, 2021.

SANDLER, M. et al. Mobilenetv2: Inverted residuals and linear bottlenecks. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 4510–4520.

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.

SHAZEER, N.; STERN, M. Adafactor: Adaptive learning rates with sublinear memory cost. In: PMLR. **International Conference on Machine Learning**. [S.l.], 2018. p. 4596–4604.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2014.

SUROWIECKI, J. **The wisdom of crowds**. [S.l.]: Anchor, 2005.

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. Inception-v4, inception-resnet and the impact of residual connections on learning. In: **Proceedings of the AAAI conference on artificial intelligence**. [S.l.: s.n.], 2017. v. 31, n. 1.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 2818–2826.

TAKAHASHI, K. et al. Confidence interval for micro-averaged f 1 and macro-averaged f 1 scores. **Applied Intelligence**, Springer, v. 52, n. 5, p. 4961–4972, 2022.

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.

VENDRAMINI, T. H. A. et al. Profile qualitative variables on the dynamics of weight loss programs in dogs. **Plos one**, Public Library of Science San Francisco, CA USA, v. 17, n. 1, p. e0261946, 2022.

WELINDER, P. et al. Caltech-ucsd birds 200. California Institute of Technology, 2010.

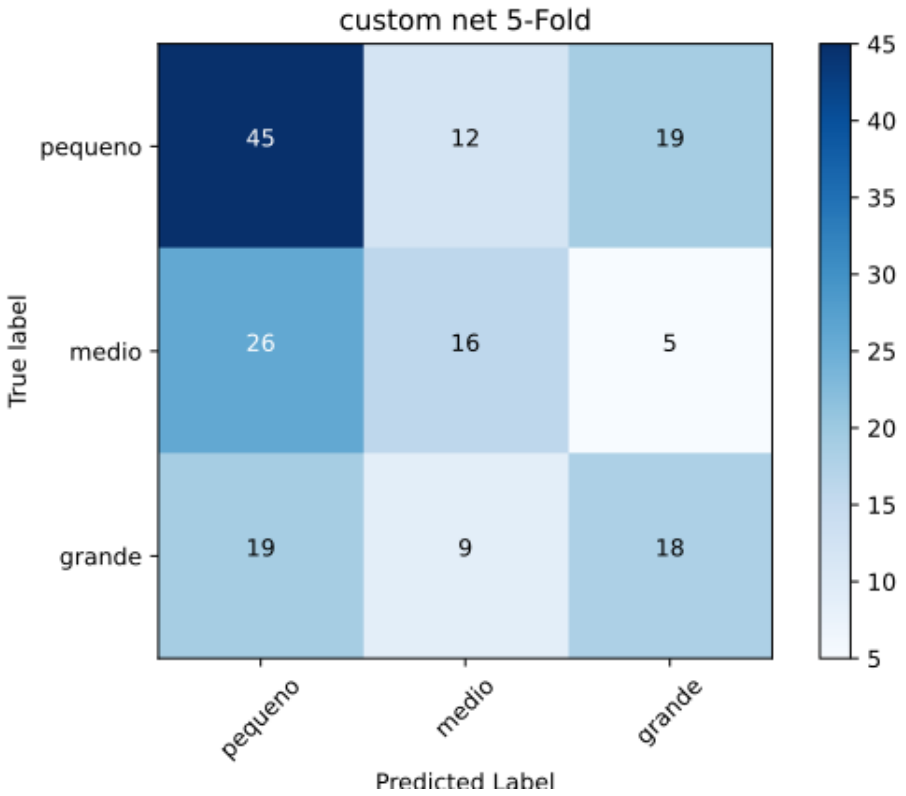
YANG, L.; SHAMI, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. **Neurocomputing**, Elsevier, v. 415, p. 295–316, 2020.

YOSINSKI, J. et al. How transferable are features in deep neural networks? **Advances in neural information processing systems**, v. 27, 2014.

ZOPH, B. et al. Learning transferable architectures for scalable image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 8697–8710.

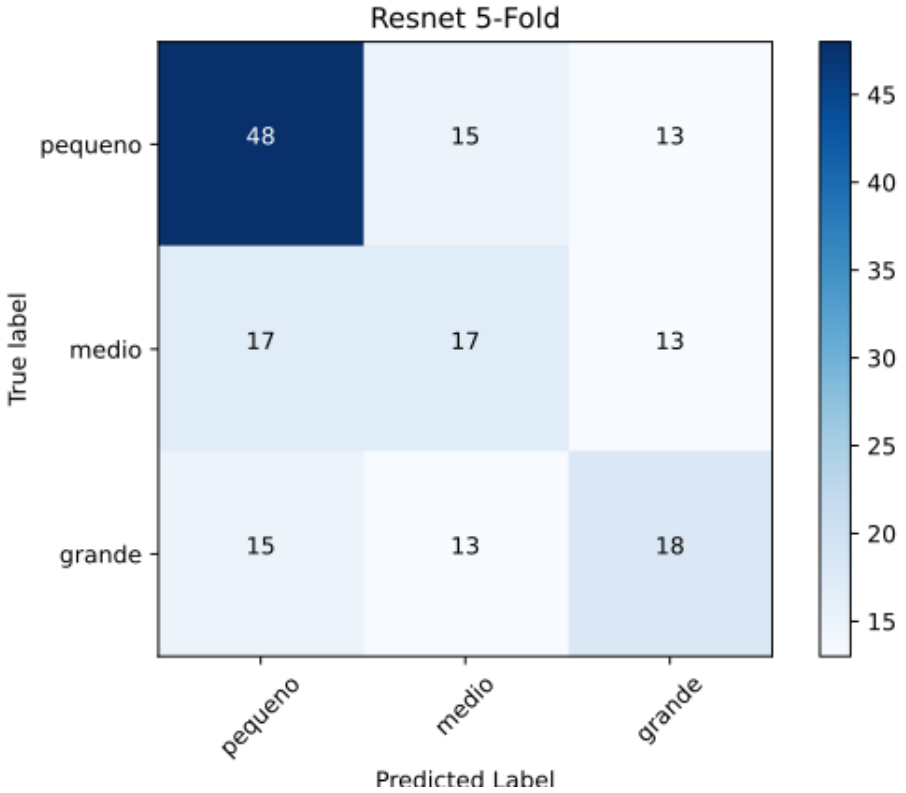
APÊNDICE A — MATRIZES DE CONFUSÃO DAS ARQUITETURAS TESTADAS

Figura A.1 – Matriz de confusão da validação cruzada na rede customizada.



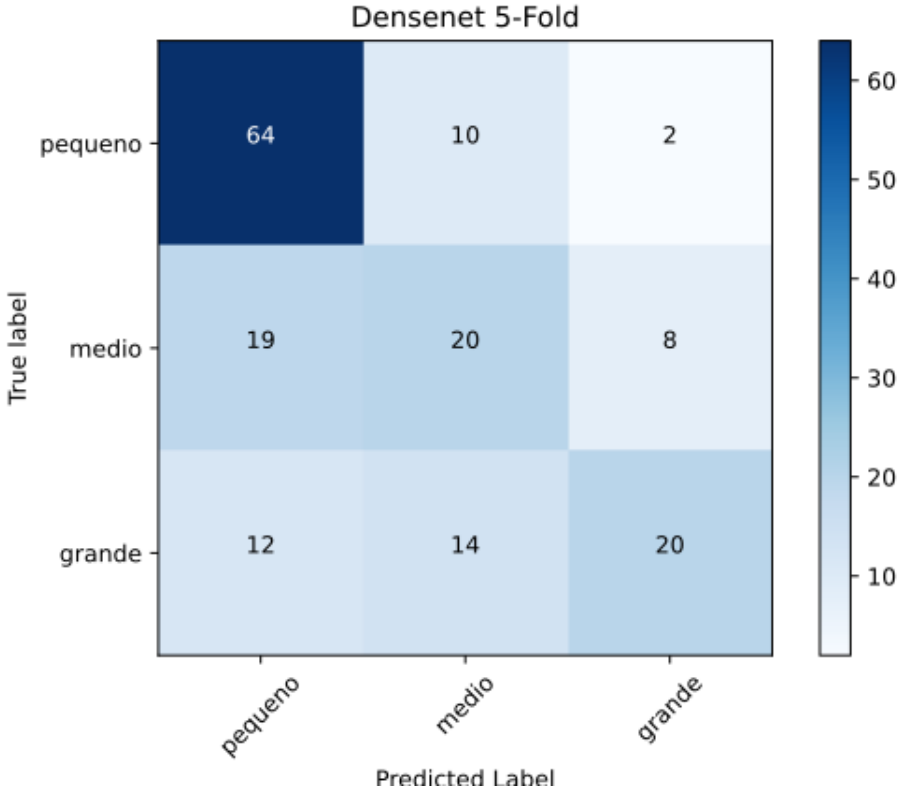
Fonte: O Autor.

Figura A.2 – Matriz de confusão da validação cruzada na ResNet.



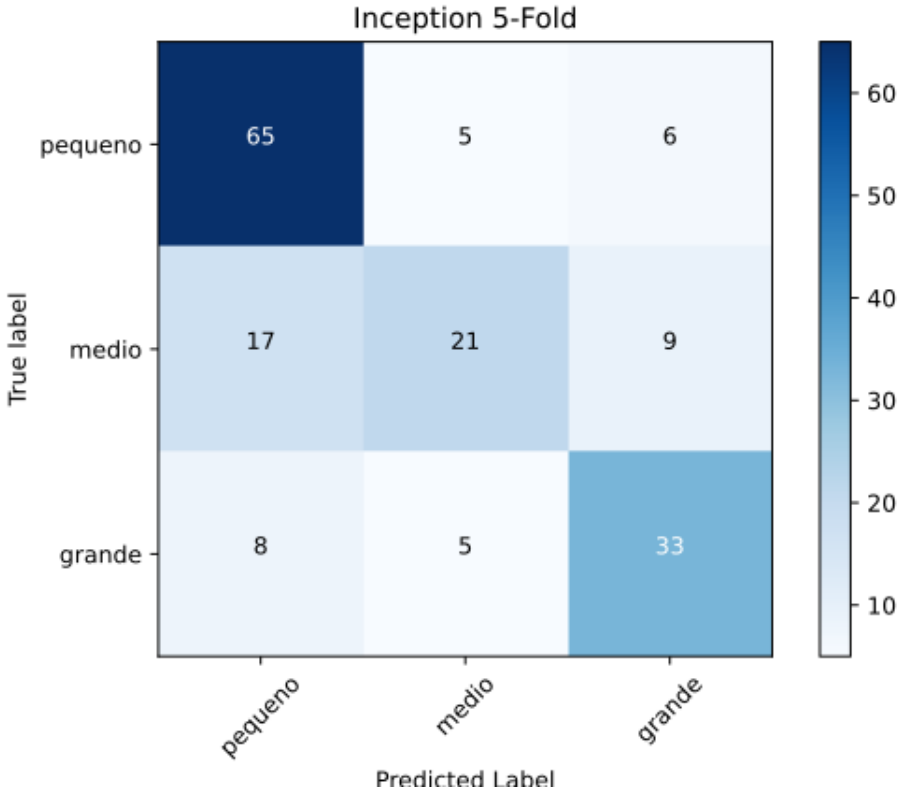
Fonte: O Autor.

Figura A.3 – Matriz de confusão da validação cruzada na DenseNet.



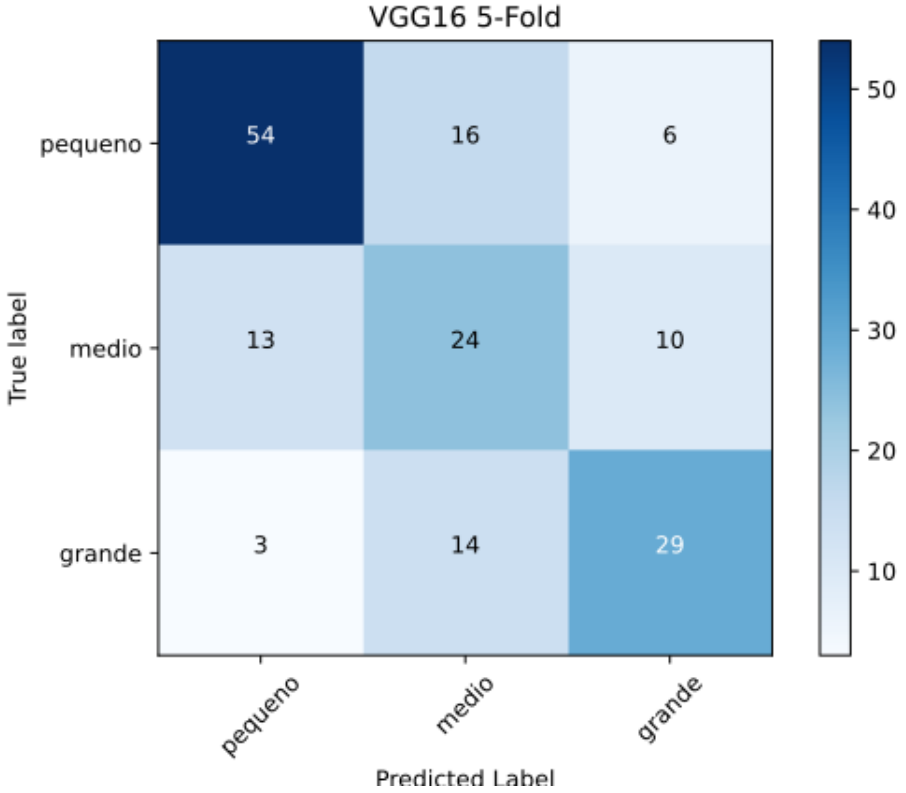
Fonte: O Autor.

Figura A.4 – Matriz de confusão da validação cruzada na Inception.



Fonte: O Autor.

Figura A.5 – Matriz de confusão da validação cruzada na VGG.



Fonte: O Autor.