UNIVERSIDADE FEDERAL DO PARANÁ

GUILHERME DE MORAES RESTANI

CLASSIFICAÇÃO DO AUMENTO DO ÁTRIO ESQUERDO EM RADIOGRAFIAS CANINAS UTILIZANDO INTELIGÊNCIA ARTIFICIAL

CURITIBA

2025

GUILHERME DE MORAES RESTANI

CLASSIFICAÇÃO DO AUMENTO DO ÁTRIO ESQUERDO EM RADIOGRAFIAS CANINAS UTILIZANDO INTELIGÊNCIA ARTIFICIAL

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Informática, no Programa de Pós-Graduação em Informática, setor de Ciências Exatas, da Universidade Federal do Paraná.

Orientador: Prof. Dr. Lucas Ferrari de Oliveira

CURITIBA

2025

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Restani, Guilherme de Moraes
Classificação do aumento do átrio esquerdo em radiografias caninas utilizando inteligência artificial / Guilherme de Moraes Restani. – Curitiba, 2025.
1 recurso on-line : PDF.

Dissertação (Mestrado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: Lucas Ferrari de Oliveira

1. Inteligência artificial - Aplicações médicas. 2. Doença Valvar Mixomatosa. 3. Cães. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Oliveira, Lucas Ferrari de. IV . Título.

Bibliotecário: Leticia Priscila Azevedo de Sousa CRB-9/2029

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **GUILHERME DE MORAES RESTANI**, intitulada: **Classificação do aumento do átrio esquerdo em radiografias caninas utilizando inteligência artificial**, sob orientação do Prof. Dr. LUCAS FERRARI DE OLIVEIRA, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.

A outorga do título de mestre está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 29 de Janeiro de 2025.

Assinatura Eletrônica
13/02/2025 14:11:23.0
LUCAS FERRARI DE OLIVEIRA
Presidente da Banca Examinadora

Assinatura Eletrônica
27/02/2025 13:26:11.0
MAUREN ABREU DE SOUZA
Avaliador Externo (PONTIFICA UNIVERSIDADE CATÓLICA DO PARANA)

Assinatura Eletrônica
13/02/2025 11:48:10.0
TILDE RODRIGUES FROES
Avaliador Externo (UNIVERSIDADE FEDERAL DO PARANÁ - UFPR)

Assinatura Eletrônica
13/02/2025 13:02:41.0
EDUARDO TODT
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

”If a machine is expected to be infallible, it cannot also be intelligent” – Alan Turing

RESUMO

A doença mixomatosa da válvula mitral (DMVM) representa cerca de 75% dos casos de doenças cardíacas diagnosticadas em cães. Ela consiste na degeneração da válvula mitral, frequentemente evoluindo para um quadro de regurgitação mitral e insuficiência cardíaca congestiva. Um dos principais elementos para seu diagnóstico é a radiografia torácica, na qual são observados elementos como a presença de aumento do átrio esquerdo. Contudo, este é um tipo de exame que exige conhecimento específico do profissional veterinário para sua interpretação, sendo que muitas vezes as clínicas veterinárias não possuem a disponibilidade deste serviço. Isto, além do grande volume de radiografias realizadas diariamente, pode resultar em diagnósticos errôneos. Sendo assim, o objetivo deste trabalho foi desenvolver um modelo capaz de detectar automaticamente o aumento do átrio esquerdo em radiografias de cães, a fim de colaborar com o diagnóstico da DMVM. Para tal, realizou-se o treinamento e análise de algoritmos de aprendizado profundo através da implementação de variados modelos de redes neurais convolucionais (CNNs) e de vision transformers (ViTs). Para treinar estas redes, confeccionou-se um dataset de radiografias torácicas latero-laterais de cães contendo pacientes normais (n=290) e com átrio esquerdo aumentado (n=160). Além das imagens originais, foram utilizadas técnicas de aumento de dados para extrapolar a quantidade de amostras. Para a avaliação dos resultados foram adotadas a metodologia de validação cruzada (5-folds) e as métricas acurácia, precisão, sensibilidade, especificidades, F-score e área sob a curva característica de operação do receptor (AUC). Para o cenário apresentado neste trabalho, as CNNs tiveram performace superior aos ViTs na maioria das métricas. O melhor F-score foi obtido pela VGG19 (0,8808 $\pm$ 0,0332) e o melhor AUC pela InceptionV3 (0,8976 $\pm$ 0,0383), ambos os casos com aumento de dados. Com estes e os demais resultados, foram construídos diferentes ensembles de modelos. Entre eles, o melhor resultado foi atingido pelo ensemble composto pelas redes InceptionV3, VGG19 e DenseNet-121, todas com aumento de dados, o qual resultou em um F-score de 0,8892 $\pm$ 0,0371 e um AUC de 0,9099 $\pm$ 0,0508. Isto é, a combinação resultou em um classificador mais equilibrado na detecção de verdadeiros positivos, minimizando os resultados falsos positivos, superando os demais modelos.

Palavras-chaves: inteligência artificial. aumento do átrio esquerdo. doença mixomatosa da válvula mitral. cão.

ABSTRACT

The myxomatous mitral valve disease (MMVD) accounts for approximately 75% of diagnosed cases of heart diseases in dogs. It consists of the degeneration of the mitral valve, often progressing to mitral regurgitation and congestive heart failure. One of the main elements for its diagnosis is chest radiography, which can assess the presence of characteristics such as the enlarged left atrium. However, this type of analysis requires specific expertise, and many veterinary clinics may lack this service. This, in combination with the high volume of X-rays generated daily, can often result in misdiagnosis. Thus, this study aims to develop a system capable of automatically detecting an enlarged left atrium in dog chest X-rays, contributing to the diagnosis of MMVD. We implemented deep learning algorithms using the artificial neural network architectures to achieve that. To train these networks, we created a dataset of lateral chest X-rays, comprising normal patients (n=290) and those with an enlarged left atrium (n=160). In addition to the original images, we employed data augmentation techniques to extrapolate the sample size. For evaluation, we adopted a 5-fold cross-validation methodology and analyzed the accuracy, precision, sensitivity, specificity, F-score, and area under the receiver operating characteristic curve (AUC) metrics. In this study, CNNs outperformed ViTs across most metrics. The highest F-score was achieved by VGG19 (0.8808 ± 0.0332), and the highest AUC by InceptionV3 (0.8976 ± 0.0383), both with data augmentation. With these and the other trained models, we constructed different ensembles. Among them, the best performance was obtained by an ensemble comprising InceptionV3, VGG19, and DenseNet-121 (all with data augmentation) resulting in an F-score of 0.8892 $\pm$ 0.0371 and an AUC of 0.9099 $\pm$ 0.0508. This combination produced a more balanced classifier for detecting true positives while minimizing false positives, surpassing the performance of all other trained models in these metrics.

Key-words: artificial intelligence. left atrial enlargement. myxomatous mitral valve disease. dog.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E DE SIGLAS

AAE Aumento do átrio esquerdo

AAE_EP Aumento do átrio esquerdo e edema pulmonar

AE Átrio esquerdo

ANN Artificial Neural Network

AUC Area Under the Curve

BCE Binary Cross-Entropy

BOF Bag-of-Features

CNN Convolutional Neural Network

CUDA Compute Unified Device Architecture

DA data augmentation

DMVM Doença mixomatosa da válvula mitral

DNN Deep Neural Network

FN Falsos Negativos

FP Falsos Positivos

GB Gigabyte

GPU Graphics processing unit

HV-UFPR Hospital Veterinário da Universidade Federal do Paraná

ICC Insuficiência cardíaca congestiva

LN Layer Normalization

LSA Locality Self-Attention

MB Megabytes

MLP Multilayer Perceptron

N Normal

NLP Natural Language Processing

RAM Random-access memory

RM Regurgitação mitral

ROC Receiver Operating Characteristic

SGD Stochastic Gradient Descent

SPT Shifted Patch Tokenization

VE Ventrículo esquerdo

VHS Vertebral Heart Scale

VLAS Vertebral Left Atrial Size

VM Válvula mitral

VN Verdadeiros Negativos

VP Verdadeiros Positivos

ViT Vision Transformer

LISTA DE SÍMBOLOS

k	um neurônio artificial qualquer
j	uma sinapse qualquer
x_j	sinal de entrada aplicado a uma sinapse j
w_{kj}	peso de uma sinapse j conectada a um neurônio k
u_k	somatório das entradas de uma sinapse ponderadas pelo peso das conexões
b_k	bias
$\varphi(.)$	função de ativação de um neurônio k
v_k	saída de um neurônio k

SUMÁRIO

1 INTRODUÇÃO

Estima-se que cerca de 10% dos cães atendidos em clínicas veterinárias sejam portadores de doenças cardíacas. Dentre elas, a mais prevalente é a doença mixomatosa da válvula mitral (DMVM), a qual representa aproximadamente 75% dos casos diagnosticados (KEENE et al., 2019).

A DMVM consiste na degeneração da válvula mitral, fazendo com que esta fique impedida de fechar corretamente. Isto implica no seu vazamento (denominado regurgitação mitral (RM)), no qual ocorre o refluxo do volume sistólico do ventrículo esquerdo para o átrio esquerdo, culminando no seu aumento e numa possível cardiomegalia. Na fase mais avançada, a DMVM pode levar à insuficiência cardíaca congestiva (ICC) (O'BRIEN; BEIJERINK; WADE, 2021), fazendo com que o coração perca sua capacidade de bombear o sangue. Aproximadamente 30% dos cães com DMVM evoluem para um quadro de regurgitação mitral e insuficiência cardíaca congestiva (PARKER; KILROY-GLYNN, 2012), sendo que na maioria destes casos o aumento do átrio esquerdo é o fator que precede o desenvolvimento da ICC (TILLEY et al., 2008).

Além disso, uma vez que a RM faz com que as altas pressões de enchimento sejam refletidas para trás, causando o aumento da pressão das veias pulmonares, ela pode culminar também no desenvolvimento de edema pulmonar (TILLEY et al., 2008).

Na maioria dos casos, a radiografia torácica é o elemento principal para o diagnóstico da DMVM, uma vez que, neste tipo de imagem, o coração e o átrio esquerdo podem ser avaliados com bom grau de confiabilidade (TILLEY et al., 2008). Este é um tipo de exame rápido, não invasivo e amplamente disponível, mas que exige conhecimento específico do profissional veterinário para sua interpretação. Dado o alto volume destes exames diariamente realizados nas clínicas veterinárias, em conjunto com o fato de que erros diagnósticos podem levar ao óbito do animal, é necessário criar mecanismos que garantam a velocidade e assertividade dos diagnósticos realizados.

Nos últimos anos, uma das áreas da ciência que apresentou evolução é a da inteligência artificial, principalmente sua sub-área do aprendizado profundo (do inglês, *deep learning*), a qual diz respeito ao desenvolvimento e implementação de algoritmos de redes neurais artificiais (do inglês, *Artificial Neural Networks* - ANNs). Os modelos computacionais gerados por estas redes são capazes de executar tarefas de classificação diretamente a partir de imagens, textos, sons, entre outros.

No campo da medicina veterinária, o uso de inteligência artificial para análise de imagens também tem se tornado popular. Uma vez que radiografias em animais são de difícil interpretação em função da variedade de espécies, o que pode ocasionar a

interpretação errônea por veterinários não especialistas em clínicas nas quais não há a presença de um radiologista (PATTARAMANEE et al., 2019), esta é uma área que vem ganhando força principalmente no que diz respeito ao desenvolvimento de sistemas que auxiliam no diagnóstico.

É possível encontrar técnicas de aprendizado profundo sendo utilizadas para diversas aplicações na medicina veterinária, como para detectar cardiomegalia em radiografias torácicas de cães (BURTI et al., 2020) e classificar radiografias torácicas de gatos (BANZATO; WODZINSKI; TAUCERI et al., 2021); para o desenvolvimento de sistemas para avaliação de grandes conjuntos de doenças em radiografias de caninos e felinos (FITZKE et al., 2021); para detectar lesões pulmonares em cães, com classificação das anormalidades (PATTARAMANEE et al., 2019); para classificação de imagem histopatológica de tumores mamários caninos (KUMAR et al., 2020); para determinar a localização de fraturas tibiais em cães e gatos (BAYDAN; BARIŞÇI; ÜNVER, 2021); para distinguir entre meningiomas e gliomas em imagens de ressonância magnética de caninos (BANZATO; BERNARDINI et al., 2018); entre outras. Isto mostra que o uso da inteligência artificial pode ser uma abordagem viável para auxiliar na análise de radiografias, podendo contribuir com o aumento da velocidade e da assertividade dos diagnósticos de doenças, como no caso da DMVM.

Usando a inteligência artificial como principal ferramenta, este trabalho busca responder às seguintes perguntas de pesquisa:

- É possível detectar o aumento do átrio esquerdo em radiografias de cães com a utilização de técnicas de aprendizado profundo?
- A utilização de técnicas para aumentar a variabilidade dos dados pode contribuir para a melhoria do desempenho dos modelos de ANNs implementados?
- Como diferentes arquiteturas de ANNs, como Redes Neurais Convolucionais (em inglês, Convolutional Neural Networks - CNN) e *Vision Transformers* (ViT), performam dado o contexto do conjunto de dados disponível?
- Ao combinar os melhores modelos gerados, é possível melhorar a assertividade geral?

1.1 OBJETIVOS

Nesta seção, são destacados os objetivos gerais e específicos que norteiam o desenvolvimento deste projeto.

1.1.1 Objetivos Gerais

Este projeto tem como objetivo geral avaliar arquiteturas de redes neurais artificiais capazes de detectar o aumento do átrio esquerdo em radiografias de cães, visando auxiliar no diagnóstico da doença mixomatosa da válvula mitral.

1.1.2 Objetivos Específicos

A lista de objetivos específicos definidos para este projeto inclui:

- Implementação e testes de diferentes modelos de aprendizado profundo para detecção do achado de interesse nas radiografias;
- Avaliação da combinação das respostas geradas pelos modelos para a melhora no resultado obtido com um único modelo;
- Aplicação de técnica de inteligência artificial explicável que permita uma compreensão visual dos modelos obtidos para análise dos resultados;
- Disponibilização dos modelos treinados para que outros pesquisadores possam refinar os modelos com seus próprios dados.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo aborda-se a doença mixomatosa da válvula mitral e a utilização das redes neurais convolucionais para a classificação de objetos em imagens.

2.1 DOENÇA MIXOMATOSA DA VÁLVULA MITRAL

A doença mixomatosa da válvula mitral (DMVM) é a doença cardíaca mais observada em cães (TILLEY et al., 2008). De acordo com O'Brien, Beijerink e Wade (2021), ela se caracteriza pelo progressivo processo de degeneração mixomatosa das válvulas atrioventriculares, especialmente do aparato da válvula mitral (VM). Na forma leve da degeneração mixomatosa mitral, observa-se a desorganização dos elementos estruturais das válvulas, juntamente com o enfraquecimento e alongamento das cordas tendíneas. A ruptura na estrutura valvar provoca a coaptação anormal dos folhetos da VM durante a sístole ventricular, permitindo o refluxo de uma parte do volume sistólico do ventrículo esquerdo (VE) para o átrio esquerdo (AE), o que é denominado regurgitação mitral (RM). Em fases mais avançadas, a fibrose secundária pode resultar na contração dos folhetos, ocasionando uma significativa deterioração da RM e uma hipertrofia excêntrica do lado esquerdo. A dilatação do coração no lado esquerdo amplifica a anormalidade na coaptação valvar, culminando na RM secundária e, por fim, na insuficiência cardíaca congestiva (ICC) do lado esquerdo.

A FIGURA 1 apresenta um esquemático da anatomia do coração canino disposta da maneira que pode ser visualizada pela projeção lateral de uma radiografia torácica. Nela é possível observar as posições relativas e proporções das estruturas do lado esquerdo do coração.

A insuficiência da válvula mitral devido à degeneração valvar pode resultar em aumento cardíaco (cardiomegalia) progressivo e a RM grave pode aumentar as pressões de enchimento do ventrículo esquerdo. Desta forma, as altas pressões de enchimento são refletidas para trás, aumentando a pressão das veias pulmonares e potencialmente iniciando o desenvolvimento de edema pulmonar.

A DMVM trata-se de uma doença adquirida, com maior prevalência na população geriátrica. Embora possa afetar qualquer raça canina, ela é observada com maior frequência nas raças de pequeno porte, como poodles miniatura, yorkshire e chihuahuas, sendo especialmente frequente em cavalier king charles spaniels, nos quais a doença pode ficar clinicamente evidente já na juventude (TILLEY et al., 2008). A DMVM pode exibir um amplo espectro de gravidade. Na maioria dos cães afetados, ela não causa sinais clínicos e acaba sendo detectada após um sopro cardíaco ser

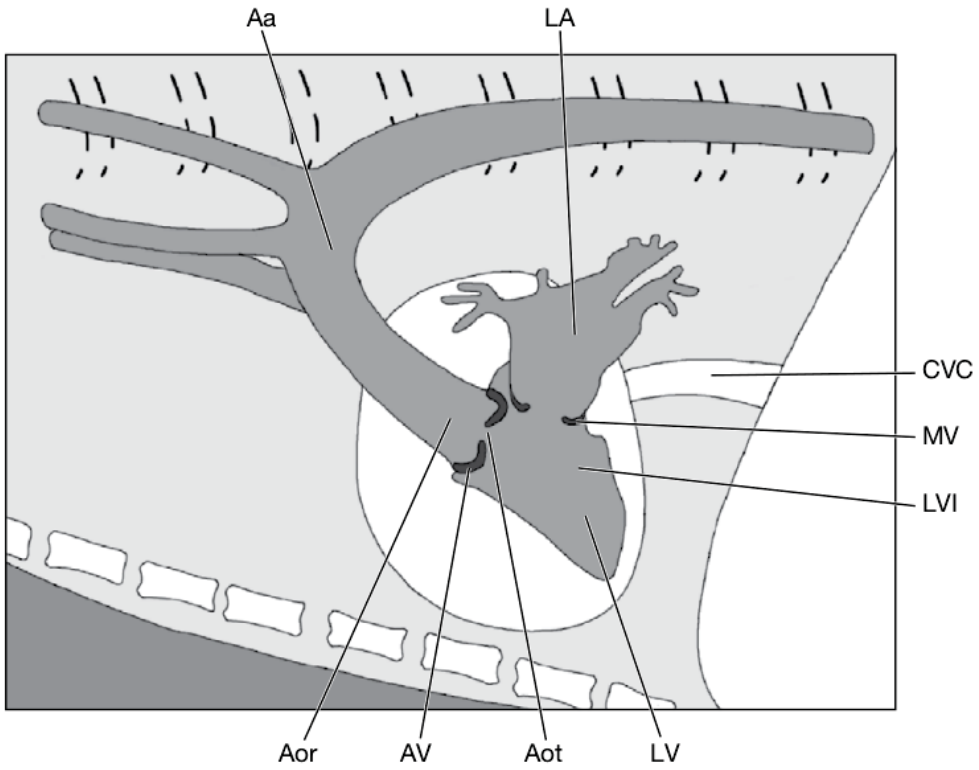


FIGURA 1 – Esquemático da projeção torácica lateral. Aa, arco aórtico; Aor, aorta; AV, válvula aórtica; Aot, via de saída aórtica; LV, ventrículo esquerdo; LVi, via de entrada do ventrículo esquerdo; LA, átrio esquerdo; MV, válvula mitral; CVC, veia cava caudal. Fonte: Tilley et al. (2008).

identificado em pacientes que se apresentam para cuidados de saúde de rotina ou para tratamento de doença não-cardíaca. Nos casos em que a DMVM se torna clinicamente aparente, a tosse é geralmente o primeiro sinal clínico observado. A tosse devido à compressão brônquica pelo átrio esquerdo aumentado costuma ser seca e forte e pode preceder o desenvolvimento de ICC. A RM pode permanecer clinicamente silenciosa até estar avançada. Quando a ICC resulta de RM, os sinais clínicos podem incluir fraqueza, síncope, tosse e dispneia (TILLEY et al., 2008).

Além disso, é importante salientar que a tosse pode estar associada à DMVM mesmo que não haja edema pulmonar. Neste caso, a tosse é um sinal de doença cardíaca, mas não de insuficiência cardíaca. Esta distinção é importante porque o diagnóstico de ICC traz importantes implicações prognósticas e terapêuticas.

Na maioria dos casos, a radiografia é o elemento mais importante para o diagnóstico da DMVM. O átrio esquerdo pode ser avaliado com bom grau de certeza neste tipo de imagem, o que é algo muito positivo, já que na grande maioria dos casos o aumento do átrio esquerdo é o fator que precede o desenvolvimento da ICC.

Uma maneira utilizada atualmente pelos veterinários como auxílio na detecção do aumento do AE com a utilização de radiografias é através do método *Vertebral Left*

Atrial Size (VLAS).

Segundo Kadotani e Fries (2022), para calcular o VLAS deve-se utilizar de projeções radiográficas laterais, nas quais mede-se a partir da margem ventral da carina até onde a borda dorsal da veia cava caudal e a silhueta cardíaca caudal se intersectam. Estas dimensões são ajustadas em relação ao comprimento das vértebras dorsais ao coração, começando a partir do aspecto cranial da quarta vértebra torácica (T4). Se a medida for superior a 2,3 (vértebras), isto pode indicar aumento do átrio esquerdo, uma vez que a escala de normalidade para cães é definida entre 1,8 e 2,3. A FIGURA 2 exemplifica a medição do VLAS.

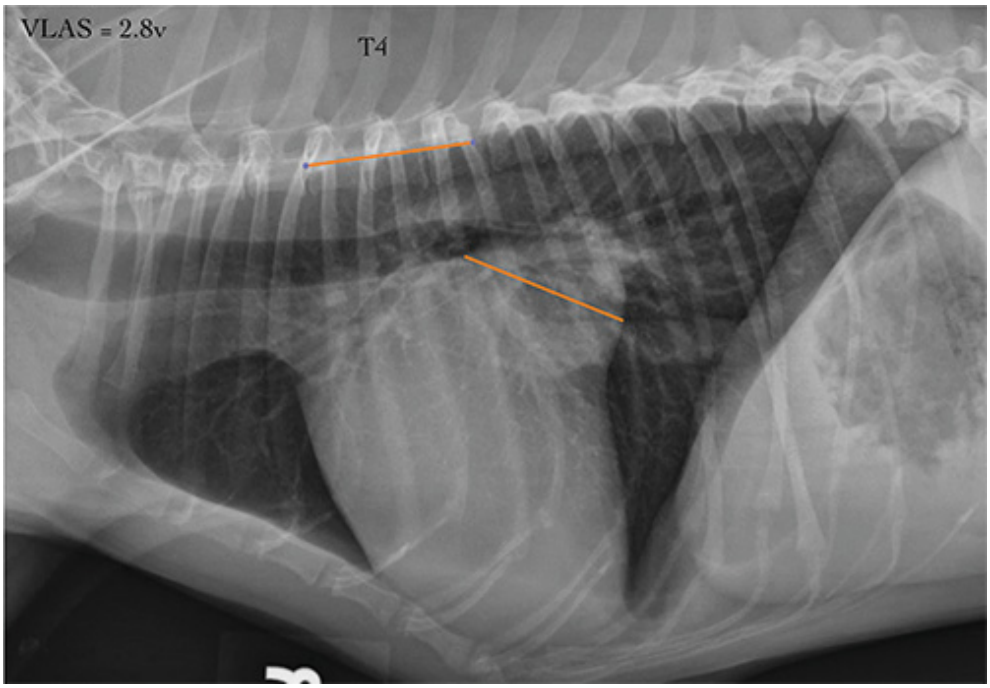


FIGURA 2 – Medição do VLAS. As linhas laranja representam a medida do AE, trasportadas para a proporção das vértebras, a partir da T4. Fonte: Kadotani e Fries (2022).

Além disso, para avaliar o tamanho do coração em radiografias torácicas, os veterinários utilizam a técnica do sistema de escala cardíaca vertebral (do inglês, Vertebral Heart Scale - VHS), a qual foi desenvolvida como um meio de avaliar objetivamente o tamanho do coração em cães de diferentes raças e diferentes conformações torácicas (ESTRADA; FOX-ALVAREZ, 2016).

Para calcular o VHS em radiografia lateral, é necessário traçar uma linha da carina até a face mais ventral do coração. Na FIGURA 3, esta linha (eixo longo) é a representada por “L”. Na sequência, é traçada uma linha, perpendicularmente a “L”, na porção mais larga do coração, estendendo-a até as bordas cranial e caudal. Na FIGURA 3, esta linha (eixo curto) é representada por “S”. Estas linhas então são transpostas, usando compassos de calibre, para a coluna vertebral, tomando como

início a face cranial da vértebra T4. Por fim, as distâncias equivalentes em quantidade de vértebras percorridas são somadas, resultando no valor do VHS (ESTRADA; FOX-ALVAREZ, 2016). Na FIGURA 3, as linhas “S”, em azul e “L”, em laranja, são transpostas para a coluna vertebral, a partir de T4, percorrendo 4,5 e 5,3 vértebras, respectivamente. Sua soma resulta em um VHS de 9,8.

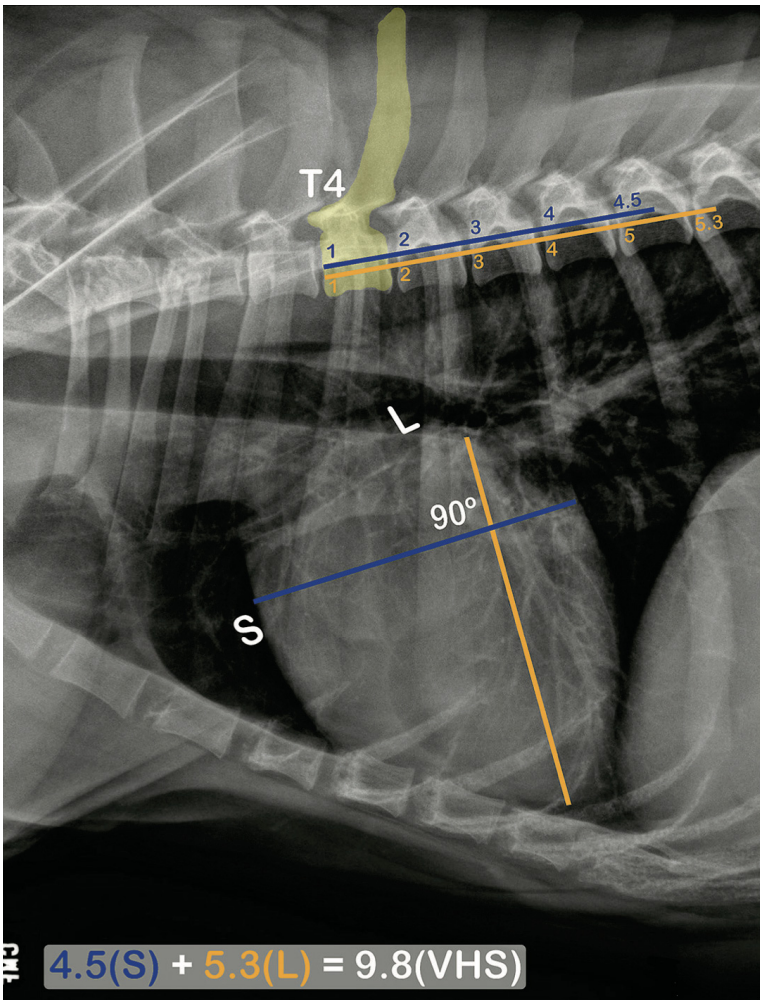


FIGURA 3 – Medição do VHS. Em laranja, linha "L", ou "eixo longo"; em azul, linha "S", ou "eixo curto". Destacado em amarelo, vértebra T4. A soma das distâncias de "L"e "S"em quantidade de vértebras resulta na medida do VHS. Fonte: Estrada e Fox-Alvarez (2016).

O intervalo de VHS considerado normal em uma radiografia lateral é de 9,2 a 10,3, com 10,5 sendo sugerido como ponto de corte para determinação clínica de cardiomegalia em cães adultos. Contudo, estes valores podem variar ligeiramente, dependendo da raça do animal (ESTRADA; FOX-ALVAREZ, 2016).

2.2 REDES NEURAIS CONVOLUCIONAIS

As Redes Neurais Artificiais, (em inglês, Artificial Neural Networks - ANN), são modelos computacionais inspirados no funcionamento das redes neuronais presentes nos seres vivos. De forma semelhante ao cérebro humano, essas redes têm a capacidade de aprender e fazer descobertas a partir de exemplos, aprimorando-se por meio da experiência. Desta maneira, este modelo computacional se torna capaz de adquirir conhecimento baseado na sua interação com conjuntos de dados, o que possibilita a realização de tarefas como a de classificar imagens, textos e sons.

Os modelos de aprendizado profundo têm sido amplamente utilizados em campos como o da visão computacional. Um dos tipos mais populares de Redes Neurais Artificiais pronfundas são as Redes Neurais Convolucionais (em inglês, Convolutional Neural Networks - CNN). Estas se destacam por sua capacidade de extrair automaticamente características dos dados de entrada através da utilização de camadas convolucionais 2D, fazendo com que esta atividade não precise ser realizada manualmente.

Nesta seção, será apresentada a teoria que aborda a estrutura e o funcionamento das Redes Neurais Artificiais com foco nas Redes Neurais do tipo convolucional.

2.2.1 Rede Neural Artificial

O cérebro humano pode ser visto como um sistema altamente complexo e não-linear de processamento de informações em paralelo. Sua habilidade de organizar seus componentes estruturais, chamados neurônios, o permite executar em alta velocidade tarefas como o reconhecimento de padrões, percepção e controle motor.

Conforme Haykin (2008), as ANNs são máquinas adaptativas que podem ser implementadas em forma de componentes eletrônicos ou de software, sendo modeladas com inspiração no comportamento cerebral. Semelhante ao cérebro, o conhecimento é adquirido a partir do meio (processo ao qual se dá o nome de aprendizado) e é armazenado através das ponderações dadas às conexões entre os neurônios.

2.2.1.1 Perceptron

O modelo matemático que implementa as características de comportamento do neurônio biológico é o perceptron (ROSENBLATT, 1958), o qual foi baseado no modelo não linear de neurônio artificial de Mcculloch e Pitts (1943). Este modelo é composto por três elementos (FIGURA 4), sendo eles um conjunto de sinapses, um somador (ou combinador linear) e uma função de ativação.

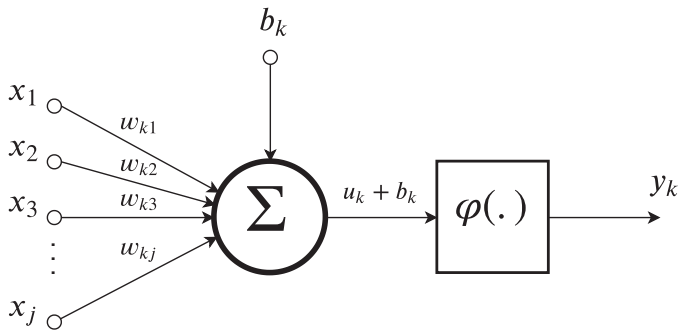


FIGURA 4 – O perceptron. Diagrama de funcionamento do perceptron. Fonte: Restani (2019).

Cada uma das sinapses do conjunto de entrada de sinais do neurônio artificial possui um peso próprio. Seja k um neurônio artificial qualquer e j uma sinapse qualquer deste neurônio, tem-se um sinal de entrada x_j que será multiplicado pelo peso desta sinapse w_{kj}.

O somador é a etapa na qual os valores obtidos pela multiplicação entre os sinais de entrada em cada sinapse pelos respectivos pesos de cada uma destas conexões são combinados. Desta forma, o somatório das entradas ponderadas u_k é dado por:

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.1}$$

Aplicado diretamente ao combinador linear do neurônio artificial há ainda o *bias* b_k, o qual é capaz de causar um deslocamento na função de ativação $\varphi(.)$.

Tomando como entrada os valores oriundos do combinador linear, a função de ativação será responsável por definir o valor final de saída do neurônio artificial. Embora um neurônio artificial possa utilizar diversas funções de ativação, para o modelo do perceptron esta será a função intitulada *threshold* (2.2), na qual **w** é o vetor dos pesos, **x** é o vetor com o conjunto de valores da entrada das sinapses e **b** é o vetor com os valores dos limiares.

$$f_{thresh}(x) = \begin{cases} 1 & \text{se} \quad \mathbf{w}^T.\mathbf{x} + b > 0, \\ 0 & \text{caso contrario.} \end{cases} \tag{2.2}$$

Dado que $u_k + b_k$ são o valores de entrada da função de ativação, tem-se que

a saída do neurônio k, dada por y_k, será:

$$y_k = \varphi(u_k + b_k) \tag{2.3}$$

Sendo então a função de ativação $\varphi(.)$ definida como a função *threshold* (2.2), um perceptron se comporta como um classificador linear binário, permitindo traçar uma fronteira entre duas classes de dados em função de suas características. Desta forma, o perceptron também pode ser utilizado para realizar operações semelhantes a portas lógicas.

2.2.1.2 Perceptron multicamadas

Em redes neurais biológicas, os neurônios frequentemente se encontram organizados em forma de camadas consecutivas (FIGURA 5). Desta mesma maneira, também é possível organizar os neurônios artificiais ao conectar os sinais de saída de um perceptron com os de entrada da camada seguinte. Este tipo de estrutura recebe o nome de perceptron multicamadas (do inglês, *Multilayer Perceptron* - MLP).

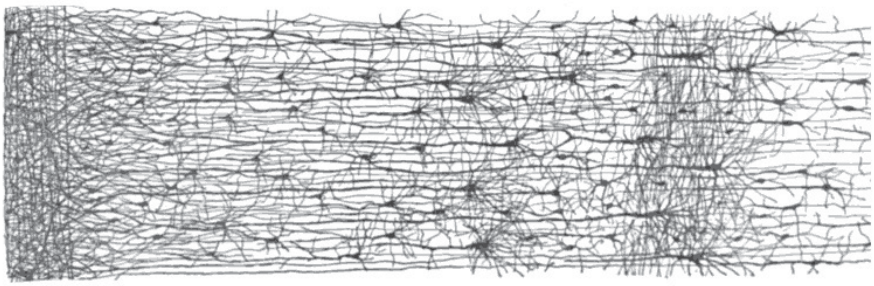


FIGURA 5 – Cortéx. Cortéx cerebral de uma criança de um ano e meio obtido com o método de Golgi. Fonte: Cajal (1899).

O MLP é dividido em três tipos de camadas de neurônios artificiais, conforme representação simplificada da FIGURA 6. Estas camadas são a camada de entrada, as camadas escondidas, e a camada de saída. O primeiro tipo de camada, a de entrada, recebe os sinais aplicados à rede. O segundo tipo, as camadas escondidas, são as quais permitem que a rede aprenda tarefas complexas por extraírem progressivamente características significativas dos padrões do vetor das entradas Haykin (2008). Por fim, tem-se a camada de saída, a qual dará o resultado final obtido em função do vetor de sinais de entrada.

Um MLP pode ter inúmeros perceptrons e também inúmeras camadas escondidas, dependendo da aplicação da rede. Ao se utilizar outros tipos de função de ativação para o neurônio artificial, é possível adicionar não-linearidade a rede, fazendo com que o MLP se torne o que se conhece por rede neural artificial. Além disso, se a estrutura

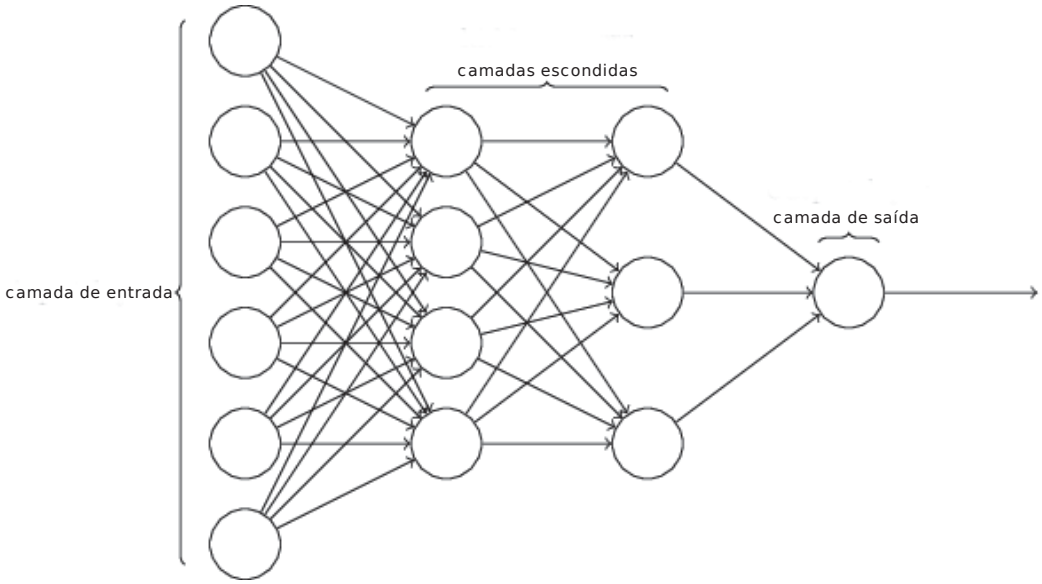


FIGURA 6 – Multilayer Perceptron. Diagrama da estrutura do MLP. Fonte: Restani (2019).

tiver duas ou mais camadas escondidas, esta é denominada rede neural profunda (do inglês, Deep Neural Network – DNN) (GÉRON, 2017).

2.2.2 Camadas das Redes Neurais Convolucionais

2.2.2.1 Convolução

A camada de convolução, a partir da qual o nome das CNNs é atribuído, é uma camada composta por um conjunto de filtros que serão convolvidos por meio de uma entrada, gerando em sua saída um mapa de características.

A FIGURA 7 exemplifica o processo de convolução[1]. Em verde temos o filtro, que é uma matriz de valores discretos de tamanho 2x2. Este filtro é convolvido, ou seja, "deslizado", através da camada de convolução, de tamanho 4x4, que é o mapa de características de uma entrada de duas dimensões. Os valores do filtro são multiplicados, elemento a elemento, com os valores da camada de convolução sobrepostos por ele. Ao final, soma-se todos os elementos resultantes, o qual irá compor o mapa de características de saída deste processo.

Neste exemplo o filtro dá um passo (ou *stride*, em inglês) unitário. Vide como o filtro, em verde, se move da FIGURA 7 (a) para FIGURA 7 (b). Mas este valor pode ser alterado. Suponha, por exemplo, um passo de tamanho 2. O filtro seria inicializado como na FIGURA 7 (a) e, na sequência, passaria direto para o estado da FIGURA 7 (c), FIGURA 7 (g) e, por fim, FIGURA 7 (i). Ao deslizar por toda a camada, o mapa de

[1] O processo exemplificado é, na verdade, a correlação cruzada e não a convolução. Para ser uma convolução o filtro primeiramente seria invertido verticalmente e depois deslizado pela camada. Contudo, estes dois termos costumam aparecer indistintamente em aprendizado de máquina.

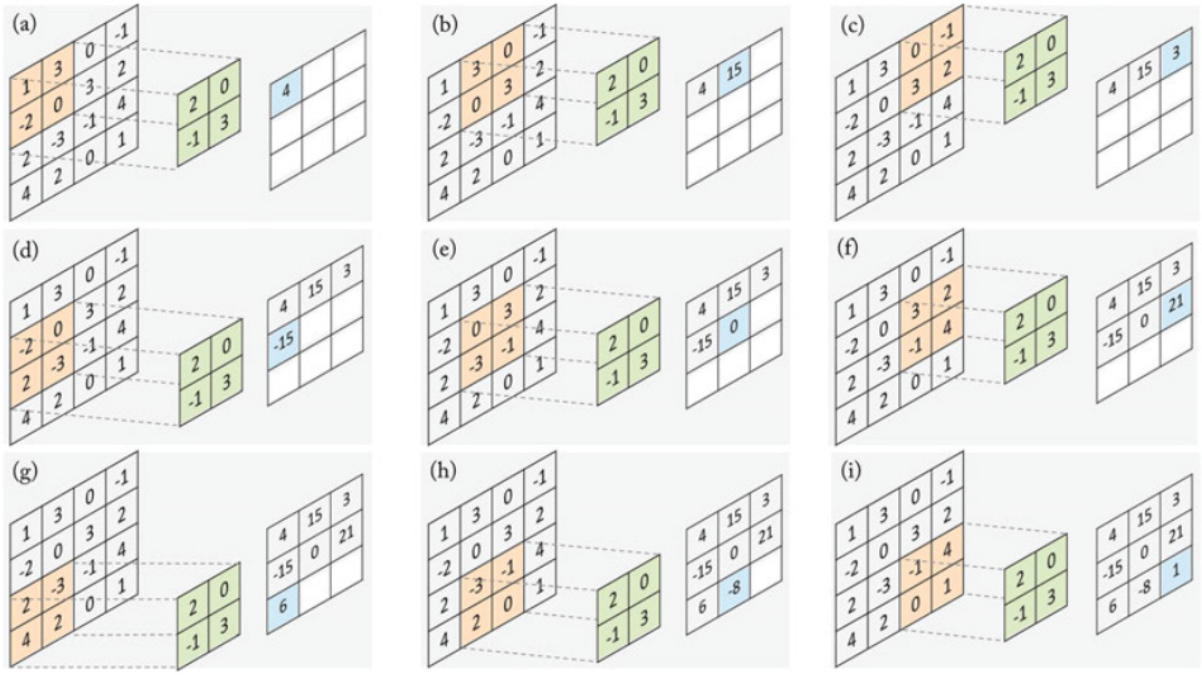


FIGURA 7 – Camada de Convolução. Em laranja: valores originais. Em verde: Filtro. Em azul: valores resultantes da operação de convolução. Fonte: Khan et al. (2018).

características resultante seria uma matriz 2x2, e não 3x3, como é o caso apresentado no exemplo para um passo de tamanho unitário.

Sendo assim, é possível observar que a dimensão do mapa de características é reduzido mantendo de certa forma a distribuição espacial dos dados. Este processo recebe o nome de operação de subamostragem (ou *sub-sampling*, do inglês). Segundo Khan et al. (2018), este processo oferece uma moderada invariância à escala e posição dos objetos, o que é uma propriedade útil em aplicações como a de reconhecimento de objetos.

Existem ainda outras maneiras de configurar o filtro, seja em dimensão ou configuração dos seus valores, para que sejam capazes de extrair características dos dados que sejam mais adequados ao tipo de aplicação, que pode ser a remoção de ruídos em imagens, segmentação, super-resolução, e assim por diante.

2.2.2.2 Pooling

Outra camada importante em uma CNN é a camada de *pooling*. Nesta camada, blocos de valores são combinados por meio de funções como média e máximo. Assim como no caso do filtro da camada de convolução, também se define o tamanho do bloco e do passo no qual a função de *pooling* será aplicada.

A FIGURA 8 exemplifica este processo. Foi definido um bloco de tamanho 2x2 e um passo de tamanho unitário. Aplica-se, então, aos valores do mapa de características na entrada da camada de *pooling* a função máximo. Na saída da camada temos como

resultado um mapa reduzido, no qual foi computado os valores máximos de cada bloco, no qual se concentra a característica com mais peso de cada região.

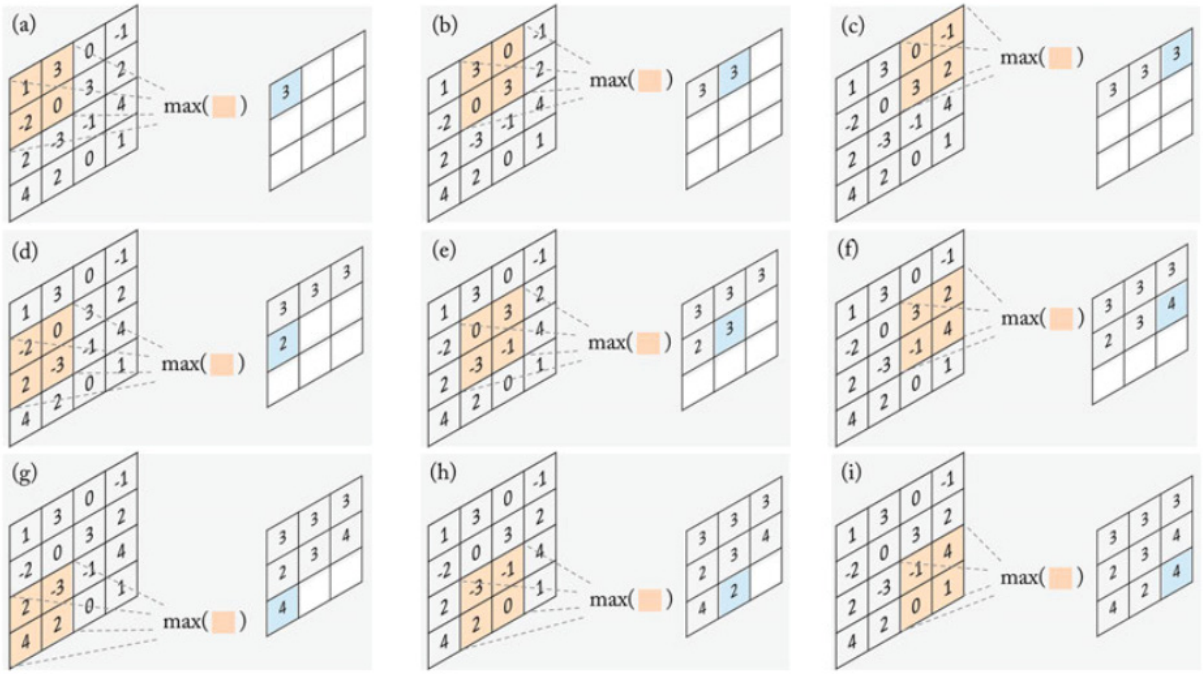


FIGURA 8 – Camada de Pooling. Em laranja: valores originais. Em azul: valor máximo da região. Fonte: Khan et al. (2018).

2.2.2.3 Camada totalmente conectada

Uma camada camada totalmente conectada (*fully connected layer*, em inglês) é essencialmente uma camada de convolução com filtros de tamanho 1x1, na qual cada unidade desta camada é densamente conectada à camada anterior (KHAN et al., 2018). Esta camada geralmente se localiza nos últimos estágios de uma CNN. Ela pode ser definida como:

$$\mathbf{y} = f(\mathbf{W}^T\mathbf{x} + \mathbf{b}) \tag{2.4}$$

onde $f(.)$ é uma função não linear, $\mathbf{x}$ é o vetor das entradas, $\mathbf{y}$ é o vetor das saídas, $\mathbf{W}$ é a matriz dos pesos das conexões entre os neurônios e $\mathbf{b}$ o vetor dos valores dos limiares. Ou seja, esta camada é análoga ao que foi visto para o MLP.

2.2.2.4 Funções de ativação

As funções de ativação dos neurônios é o modo pelo qual é possível inserir não-linearidade à rede, permitindo que eles se comportem de maneira mais próxima ao funcionamento do neurônio biológico. As funções de ativação tomam um valor de entrada real e o comprime em um intervalo geralmente como [0, 1] ou [-1, 1].

Uma função não linear pode ser entendida como um mecanismo de comutação ou de seleção, o qual irá decidir se um neurônio se ativa ou não dados todos os seus valores de entrada (KHAN et al., 2018).

A seguir serão apresentadas as funções de ativação que aparecem com mais frequência no escopo das ANNs.

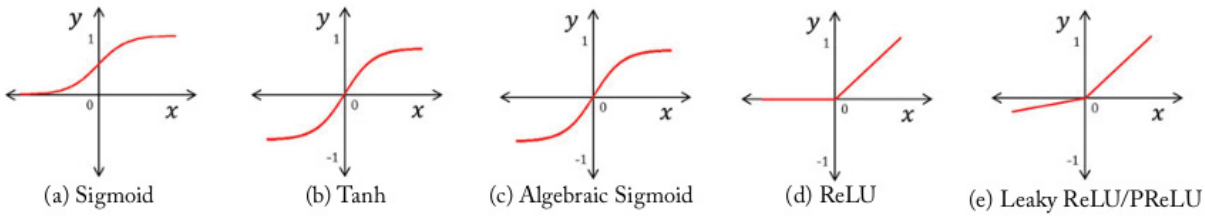


FIGURA 9 – Funções de Ativação. Funções responsáveis por adicionar não-linearidade à ANN. Fonte: Adaptado de Khan et al. (2018).

Sigmoid – A função de ativação sigmoidal (FIGURA 9 a) irá receber um valor real em sua entrada e apresentar uma saída no intervalo [0, 1].

$$f_{sigm}(x) = \frac{1}{1 + e^{-x}} \tag{2.5}$$

Tanh – A função de ativação tangente hiperbólica (FIGURA 9 b) é muito semelhante à função sigmoidal, ela irá receber um valor real em sua entrada porém apresentar uma saída no intervalo [1, -1].

$$f_{tanh}(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.6}$$

Algebraic Sigmoid Function – A função de ativação sigmoidal algébrica (FIGURA 9 c) é muito semelhante à função tangente hiperbólica, sendo que sua diferença está apenas no grau de inclinação da curva.

$$f_{a-sig}(x) = \frac{x}{\sqrt{1 + x^{2}}} \tag{2.7}$$

Rectifier Linear Unit (ReLu) – Baseada no processamento do cortex visual humano (HAHNIOSER et al., 2000), a ReLu (FIGURA 9 d) é uma das funções de ativação com maior importância prática. Ela mapeia o valor de entrada do neurônio para 0, caso este valor seja negativo, e mantém o valor original, caso este seja positivo. Dada esta característica, a ReLu é computada em grande velocidade.

$$f_{relu}(x) = max(0, x) \tag{2.8}$$

Leaky ReLu – Dado o sucesso da ReLu, muitas variantes surgem dela. Um exemplo é a Leaky Relu (FIGURA 9 d), que mantém o sinal de saída caso a entrada seja positiva e reduz a escala do valor caso ele seja negativo.

$$f_{l-rel}(x) = \begin{cases} x & \text{se } \ x > 0 \\ cx & \text{se } x \leq 0 \end{cases} \tag{2.9}$$

2.2.3 Arquiteturas de Redes Neurais Convolucionais

Nesta seção são descritas as arquiteturas de CNNs escolhidas para serem utilizadas neste trabalho.

2.2.3.1 VGG

A arquitetura VGG é o resultado de uma investigação realizada por Simonyan e Zisserman (2015) sobre o efeito que a profundidade das redes neurais convolucionais tem em sua acurácia para cenários de reconhecimento de imagens em conjuntos de dados em larga escala.

Para tal, foram utilizados filtros convolucionais pequenos (3×3) em todas as camadas, enquanto os demais parâmetros foram mantidos fixos. Progressivamente, mais camadas convolucionais foram sendo adicionadas. Desta forma, os autores chegaram a duas configurações com melhor desempenho, sendo elas com 16 e 19 camadas convolucionais.

A VGG toma como entrada da rede uma imagem RGB de tamanho fixo com 224×224 pixels, a qual é processada por uma sequência de camadas convolucionais. Nestas camadas são utilizados filtros com um campo receptivo de 3×3, o que é suficiente para capturar direções como esquerda/direita, cima/baixo e centro. O deslocamento (stride) da convolução é fixado em 1 pixel, e o preenchimento espacial (padding) é ajustado de forma que a resolução espacial seja preservada após a convolução. Todas as camadas ocultas da rede utilizam a função de ativação não-linear ReLU (rectified linear unit).

O agrupamento espacial é realizado por cinco camadas de max-pooling, que seguem algumas das camadas convolucionais (nem todas são seguidas por max-pooling). O max-pooling é executado em uma janela de 2×2 pixels, com deslocamento de 2 pixels.

Após a sequência de camadas convolucionais, a rede é seguida por três camadas totalmente conectadas (Fully-Connected– FC). As duas primeiras camadas totalmente conectadas tem 4096 neurônios cada, e a terceira tem 1000 neurônios, uma vez que foi projetada para realizar a classificação no conjunto de dados ILSVRC15

(RUSSAKOVSKY et al., 2015), o qual é composto por 1000 classes. Por fim, na última camada é aplicada a função softmax.

As arquiteturas das redes VGG-16 e VGG-19 são apresentadas na FIGURA 10 e na FIGURA 11, respectivamente. Nelas é possível observar como a quantidade de camadas intermediárias diferem entre as duas arquiteturas, contudo mantendo a mesma quantidade de camadas de max-pooling.

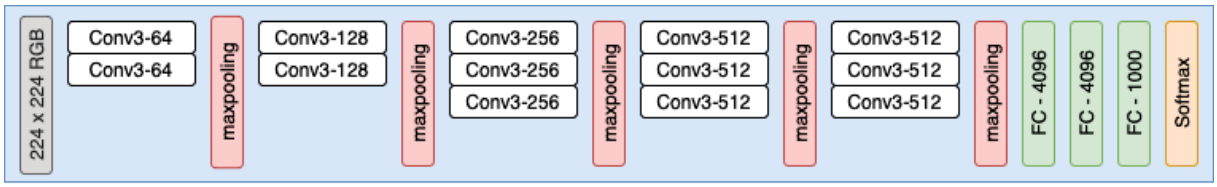


FIGURA 10 – VGG16. Em cinza, imagem de entrada. Em branco, 13 camadas de convolução. Em vermelho, camadas de max-pooling que seguem as camadas de convolução. Em verde, 3 camadas totalmente conectadas. Em amarelo, função softmax. Fonte: O autor (2024).

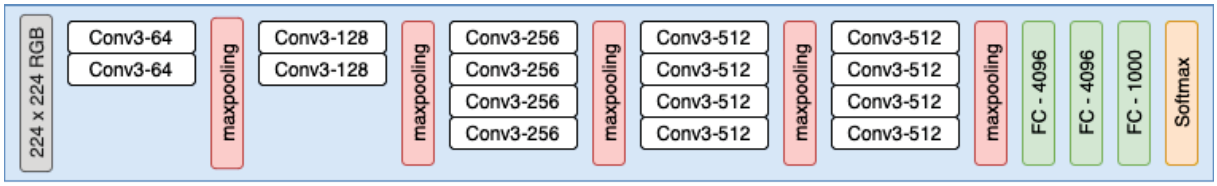


FIGURA 11 – VGG19. Em cinza, imagem de entrada. Em branco, 16 camadas de convolução. Em vermelho, camadas de max-pooling que seguem as camadas de convolução. Em verde, 3 camadas totalmente conectadas. Em amarelo, função softmax. Fonte: O autor (2024).

2.2.3.2 ResNetV2

Ao ficarem cada vez mais complexas e profundas, uma questão sobre o aprendizado nas redes neurais artificias que ficou evidente é o problema do vanishing/exploding gradient, o qual ocorre quando os gradientes ficam extremamente pequenos durante a retropropagação em redes neurais profundas, dificultando o aprendizado das camadas iniciais. Este problema fazia com que a acurácia saturasse e degradasse rapidamente, dificultando a convergência da rede desde o início.

Uma família de redes neurais que surgiram para endereçar este problema é a das redes residuais profundas (do inglês, deep residual networks), também conhecidas como ResNets (HE et al., 2015). Estas são redes extremamente profundas, mas capazes de mostrar bons resultados de acurácia e de comportamentos de convergência.

As ResNets tem como unidade fundamental o chamado bloco residual. O qual é definido na EQUAÇÃ0 2.10, onde **x** e **y** são os vetores de entrada e saída,

respectivamente, das camadas da rede, e Wi são os pesos dos neurônios da camada.

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x} \tag{2.10}$$

Isto é, $\mathcal{F}(x) + x$ pode ser vista como uma espécie de atalho entre as conexões da rede, que aplicam uma função de mapeamento identidade. O que, por sua vez, não adiciona parâmetros ou complexidade computacional extras.

Em resumo, nas redes neurais tradicionais, a saída de cada camada alimenta a camada seguinte. Contudo, no caso das ResNets, a saída de uma camada irá alimentar a camada imediatamente a seguir, mas também realizará um salto, alimentando camadas mais distantes. O diagrama do funcionamento do bloco residual pode ser verificado na FIGURA 12, a qual exemplifica este processo.

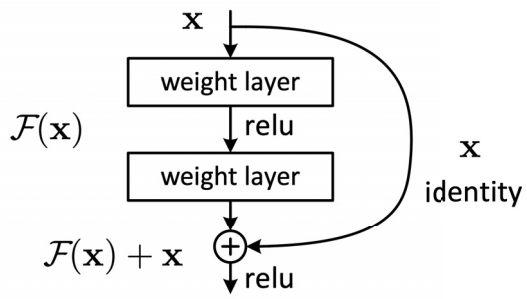


FIGURA 12 – Bloco residual. Diagrama do bloco residual. Fonte: He et al. (2015).

Ao analizar as formulações de propagação de informação nos blocos residuais, os autores da ResNet perceberam que estas sugerem que sinais podem ser propagados e retropropagados de um bloco para qualquer outro, ao usar a função de mapeamentos de identidade como salto entre conexões e como ativação pós-adição.

Desta forma, melhoraram a arquitetura da ResNet ao focarem na criação de um caminho “direto” para a propagação de informação não só entre as unidades residuais, mas através de toda a rede (HE et al., 2016). Esta arquitetura ficou conhecida como ResNetV2. A FIGURA 13 mostra o caminho original da informação na ResNet (à esquerda) em comparação com o novo caminho proposto (à direita), na qual a seta em cinza indica o caminho mais fácil para a propagação da informação. É possível observar que no caminho proposto as funções de ativação ReLU e de normalização em lote (do inglês, batch normalization - BN (IOFFE; SZEGEDY, 2015)) atuam como “pré-ativação” das camadas de pesos, em contraste com a implementação mais convencional de “pós-ativação”.

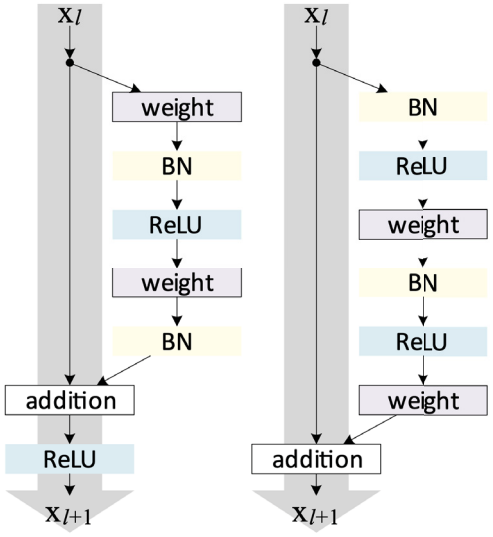


FIGURA 13 – Caminho da informação na ResNetV2. Caminho original da informação na ResNet (à esquerda) versus novo caminho proposto (à direita) para a ResNetV2. Fonte: He et al. (2015).

2.2.3.3 InceptionV3

Baseada na rede Inception (SZEGEDY; LIU et al., 2014), originalmente desenvolvida para tarefas de visão computacional, a InceptionV3 (SZEGEDY; VANHOUCKE et al., 2016) é uma versão aprimorada desta arquitetura. Foram várias as modificações implementadas pelos autores no modelo original, as quais se mostraram capazes de aprimorar a eficiência e o desempenho do algoritmo.

As principais características da InceptionV3 são: Fatorização em Convoluções Menores, Fatorização Espacial em Convoluções Assimétricas, Classificadores Auxiliares, e Redução Eficiente do Tamanho do Grid.

Convoluções com filtros espaciais grandes (por exemplo, 5x5 ou 7x7), tendem a ser computacionalmente caras. Contudo, um filtro maior pode capturar dependências entre sinais entre ativações de unidades mais distantes nas camadas anteriores, ou seja, uma redução do tamanho geométrico dos filtros resulta em um custo de expressividade. Para melhorar o desempenho da rede, os autores utilizaram a fatorização (ou decomposição) dos filtros grandes em convoluções menores, substituindo-os por redes multicamadas com menos parâmetros e com o mesmo tamanho de entrada e profundidade de saída. A FIGURA 14 mostra a estrutura original, à esquerda, e a nova estrutura proposta, à direita, na qual o filtro de 5x5 é fatorado em duas camadas de filtros 3x3.

Uma vez que este resultado sugere que convoluções com filtros maiores do que 3x3 não seriam úteis, já que estas poderiam ser reduzidas em sequências de camadas de filtros convolucionais 3x3, os autores investigaram o ganho em fatorá-las

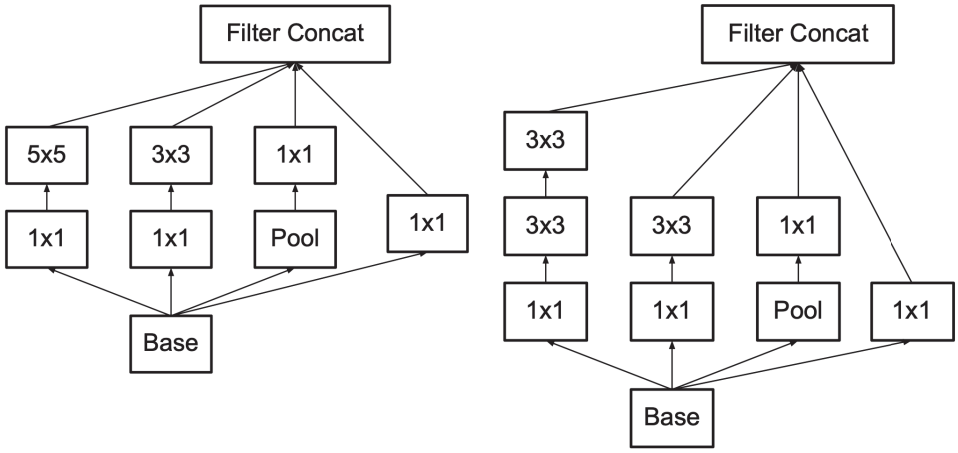


FIGURA 14 – Fatorização em filtros menores. Estrutura original da Inception, à esquerda, e a nova estrutura proposta, à direita, na qual o filtro de 5x5 é fatorado em duas camadas de filtros 3x3. Fonte: Szegedy, Vanhoucke et al. (2016).

em convoluções ainda menores. Perceberam, então, que poderiam utilizar convoluções assimétricas, e.g. nx1. Por exemplo, usar uma convolução 3×1 seguida por uma convolução 1×3 é equivalente a deslizar uma rede de duas camadas com a mesma convolução 3×3. Além disso, mesmo com duas camadas, esta solução se mostrou 33% mais barata computacionalmente.

Em (SZEGEDY; LIU et al., 2014) foi introduzida a noção de que classificadores auxiliares melhoram a convergência de redes neurais muito profundas. Contudo, Szegedy, Vanhoucke et al. (2016) descobriram que os classificadores auxiliares não melhoravam a convergência dos modelos nos estágios iniciais dos treinamentos. Contudo, perto do final do treinamento, a rede com os ramais auxiliares começa a ultrapassar a acurácia da rede sem nenhum ramal auxiliar, atingindo um plateau final maior. Desta forma, os classificadores auxiliares atuariam como regularizadores. Sendo assim, na arquitetura proposta estes classificadores foram reorganizados, sendo removidos dos estágios iniciais da rede.

Tradicionalmente, as redes convolucionais usavam algumas operações de pooling para diminuir o tamanho do grid dos mapas de características. Para evitar um gargalo representacional, antes de aplicar o pooling máximo ou médio, a dimensão de ativação dos filtros da rede é expandida. Aqui Szegedy, Vanhoucke et al. (2016) sugerem uma outra variante que reduz ainda mais o custo computacional, ao mesmo tempo que remove o gargalo representacional. Esta configuração pode ser visualizada na FIGURA 15, na qual são implementados dois blocos paralelos de stride 2: P e C. No qual P é uma camada de pooling (pooling médio ou máximo) de ativação, sendo que ambos são stride 2 cujos bancos de filtros estão concatenados.

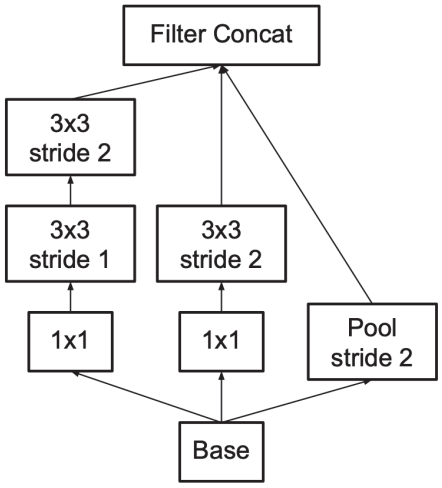


FIGURA 15 – Redução Eficiente do Tamanho do Grid. Estratégia de pooling utilizada na InceptionV3 para reduzir o mapa de características sem criar gargalos na estrutura da rede. Fonte: Szegedy, Vanhoucke et al. (2016).

A estrutura geral da arquitetura InceptionV3 pode ser visualizada de maneira simplificada na FIGURA 16, a seguir.

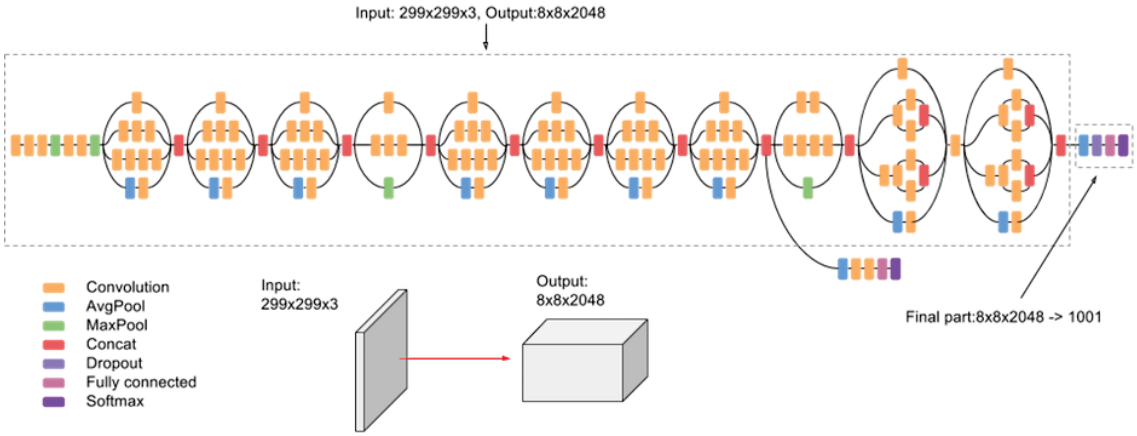


FIGURA 16 – Diagrama da rede InceptionV3. Diagrama simplificado ilustrativo da arquitetura de rede InceptionV3. Fonte: Google (2023).

2.2.3.4 Xception

A arquitetura Xception (CHOLLET, 2017), cujo nome deriva de "Extreme Inception", é uma rede neural convolucional profunda inspirada na arquitetura Inception, na qual os módulos Inception foram completamente substituídos por camadas de convolução separáveis em profundidade (do original em inglês, "depthwise separable convolution"). A hipótese subjacente à sua concepção é de que o mapeamento das correlações entre canais e das correlações espaciais nos mapas de características de redes neurais convolucionais pode ser inteiramente desacoplado. Desta forma,

reduz-se significativamente o número de parâmetros e o custo computacional, porém mantendo o poder de representação da rede.

As convoluções separáveis em profundidade diferem das convoluções tradicionais, as quais operam simultaneamente sobre as dimensões espaciais e de profundidade, ao realizar essas operações de forma independente.

A arquitetura Xception possui 36 camadas convolucionais que formam a base de extração de características da rede. Para o caso da tarefa de classificação de imagens, a base composta pelas convoluções é seguida por uma camada de regressão logística. Essas 36 camadas convolucionais estão organizadas em 14 módulos, todos com conexões residuais lineares ao seu redor (assim como na ResNet), exceto os primeiros e últimos módulos.

Na FIGURA 17, está representada esta estrutura da Xception, a qual consiste em uma pilha linear de camadas de convolução separável em profundidade com conexões residuais. O fluxo de dados na arquitetura inicia-se no “entry flow”, seguido pelo “middle flow”, que é repetido oito vezes, e finaliza no “exit flow”. Embora não esteja representado no diagrama, todas as camadas de convolução e convolução separáveis são seguidas de normalização em lote (batch normalization). Além disso, todas as camadas de convolução separáveis utilizam um multiplicador de profundidade de 1, sem expansão de profundidade.

2.2.3.5 DenseNet121

Proposta por Huang et al. (2017), a Rede Densa Convolucional (do inglês, Dense Convolutional Network – DenseNet) é uma arquitetura que implementa um padrão de conectividade com o objetivo de assegurar o fluxo máximo de informações entre suas camadas. A fim de manter a natureza de propagação direta (feed-forward), cada camada recebe entradas adicionais de todas as camadas precedentes e transmite seus próprios mapas de características a todas as camadas subsequentes. Esse arranjo é ilustrado de forma esquemática na FIGURA 18.

Desta forma, o mapa de características resultante de cada camada é utilizado como entrada por todas as camadas subsequentes a ela, e assim sucessivamente. Como resultado deste fluxo da informação, a DenseNet oferece vantagens como a mitigação do problema do gradiente em dissipação (vanishing gradient), o reforço da propagação de características, além de compelir a reutilização das características e promover a redução no número de parâmetros.

Diferentemente das redes residuais (ResNets), as quais também tratam o problema do gradiente em dissipação, no caso da DenseNet as características não são combinadas por meio de soma, mas por concatenação, antes de serem passadas para

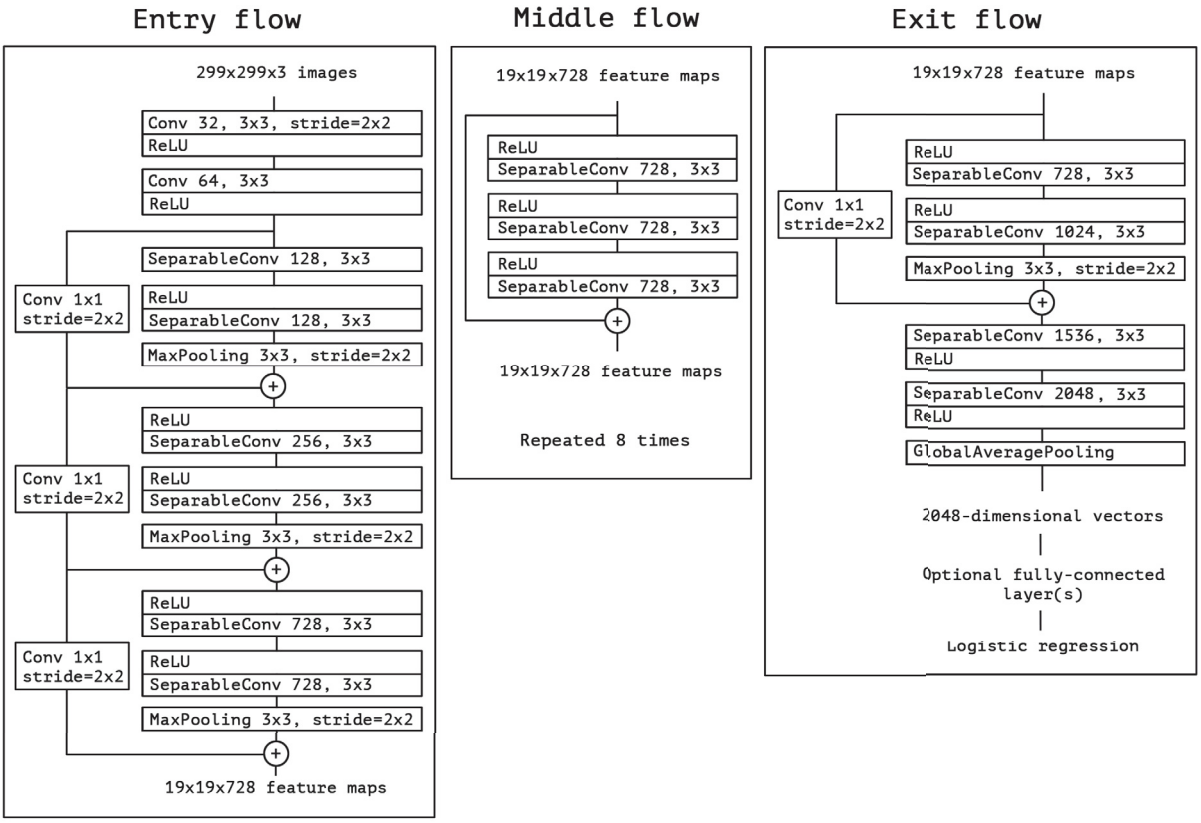


FIGURA 17 – Diagrama da rede Xception. Detalhe dos fluxos de entrada (entry flow), intermediário (middle flow) e de saída (exit flow). Fonte: Chollet (2017).

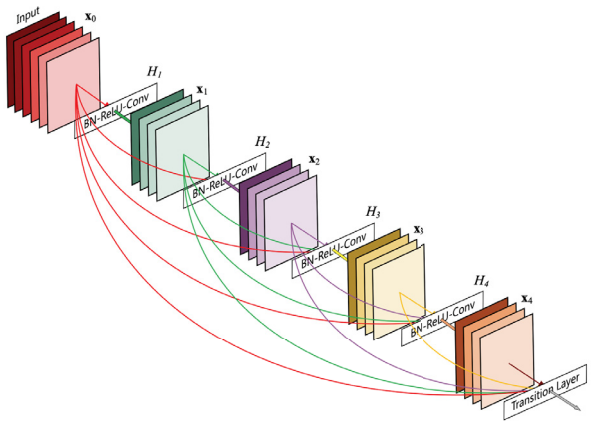


FIGURA 18 – DenseNet. Arranjo da rede DenseNet destacando como o mapa de características resultante de uma camada é utilizado como entrada por todas as camadas subsequentes. Fonte: Huang et al. (2017).

uma camada. Sendo assim, uma camada l possui l entradas, consistindo nos mapas de características de todos os blocos convolucionais anteriores. Seus próprios mapas de características são transmitidos para todas as $L - l$ camadas subsequentes, resultando em $L(L + 1)/2$ conexões em uma rede de L camadas - em vez de apenas L, como nas arquiteturas convencionais. A FIGURA 19 apresenta a estrutura de uma DenseNet com 121 camadas. Ao final, é possível observar uma camada de classificação com 1000

neurônios, uma vez que esta foi desenvolvida para realizar a classificação das 1000 classes do conjunto de dados ImageNet (DENG et al., 2009).

Layers	Output Size	DenseNet-121
Convolution	112×112	7×7 conv, stride 2
Pooling	56×56	3×3 max pool, stride 2
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv
	28×28	2×2 average pool, stride 2
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv
	14×14	2×2 average pool, stride 2
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Transition Layer (3)	14×14	1×1 conv
	7×7	2×2 average pool, stride 2
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$
Classification Layer	1×1	7×7 global average pool
		1000D fully-connected, softmax

FIGURA 19 – DenseNet-121. Estrutura de uma DenseNet com 121 camadas. Fonte: Huang et al. (2017).

Embora possa parecer contraintuitivo, este padrão de conectividade densa demanda uma menor quantidade de parâmetros em comparação com as redes convolucionais tradicionais, uma vez que não há necessidade do reaprendizado de mapas de características redundantes. Além disso, outra vantagem das DenseNets é possibilitar um fluxo aprimorado de informações e gradientes ao longo da rede, o que facilita o seu treinamento. Cada camada possui acesso direto aos gradientes da função de perda e ao sinal de entrada original, acarretando em uma supervisão profunda implícita. Isto favorece o treinamento de arquiteturas de redes mais profundas. Por fim, ainda observa-se que as conexões densas exercem um efeito de regularização, reduzindo o *overfitting* em conjuntos de treinamento menores.

2.2.3.6 NASNet

A arquitetura NASNet (ZOPH; VASUDEVAN et al., 2018) é inspirada no *framework* do Neural Architecture Search (NAS) proposto por Zoph e Le (2017), o qual se baseia em um método de busca por aprendizado por reforço para otimizar configurações de arquitetura de rede. Como a aplicação do NAS em grandes conjuntos de dados é computacionalmente custosa, a NASNet propõe buscar uma arquitetura

eficiente em um conjunto de dados substituto e menor, e então transferir a arquitetura aprendida para um conjunto maior.

No NAS, uma rede neural recorrente (RNN) de controle amostra redes ”filhas” com diferentes arquiteturas. Estas redes ”filhas” são treinadas até a convergência para alcançar uma dada acurácia. As acurácias obtidas atualizam o controlador, de modo que ele gere arquiteturas progressivamente melhores. Já a NASNet, para simplificar o processo de busca e construção da rede, a busca pela melhor arquitetura convolucional é reduzida à busca pela melhor estrutura de célula. Como todas as redes utilizadas são compostas por camadas convolucionais com estrutura idêntica, mas pesos diferentes, o espaço de busca da NASNet é construído visando alcançar transferibilidade do modelo construído a partir de um conjunto de dados reduzido para um maior.

As duas unidades principais (ou células) que a busca se concentra em encontrar são a célula normal (normal cell) e a célula de redução (reduction cell). A primeira se trata de células convolucionais que retornam um mapa de características com a mesma dimensão, e a segunda se trata de células convolucionais que retornam um mapa de características cuja altura e largura são reduzidas pela metade, o que ajuda a comprimir informações e a capturar características mais abstratas nas camadas mais profundas da rede. Por fim, as células encontradas são dispostas em blocos repetitivos ao longo da rede, criando uma estrutura hierárquica. As células ”normais” e de ”redução” são combinadas para formar camadas em diferentes estágios da rede. A FIGURA 20 ilustra a disposição das células normais e de redução para os datasets CIFAR-10 e ImageNet. No caso do ImageNet a arquitetura é composta por mais células de redução, uma vez que o tamanho da imagem de entrada é 299x299, em comparação com 32x32 do CIFAR. O que varia nas redes convolucionais são as estruturas das células normais e de redução, as quais são determinadas pela RNN controladora.

Esta abordagem com foco na busca por estruturas de célula apresenta duas vantagens principais. A primeira é ser muito mais rápida do que buscar por uma arquitetura completa de rede e a segunda é o fato de que a própria célula tem maior probabilidade de generalizar para outros problemas. Além disso, ao variar o número de células convolucionais e o número de filtros nas células convolucionais, é possível criar diferentes versões de NASNets, para distintas demandas computacionais, o que a torna escalável.

2.2.4 Comparativo entre os modelos de CNNs

A seguir, na TABELA 1, estão compiladas características das CNNs descritas anteriormente, como os seus tamanhos em Megabytes (MB), as suas quantidades de parâmetros (Params), as suas quantidades de camadas e as suas acurácias (Acc), para modelos implementados para o conjunto de dados ImageNet.

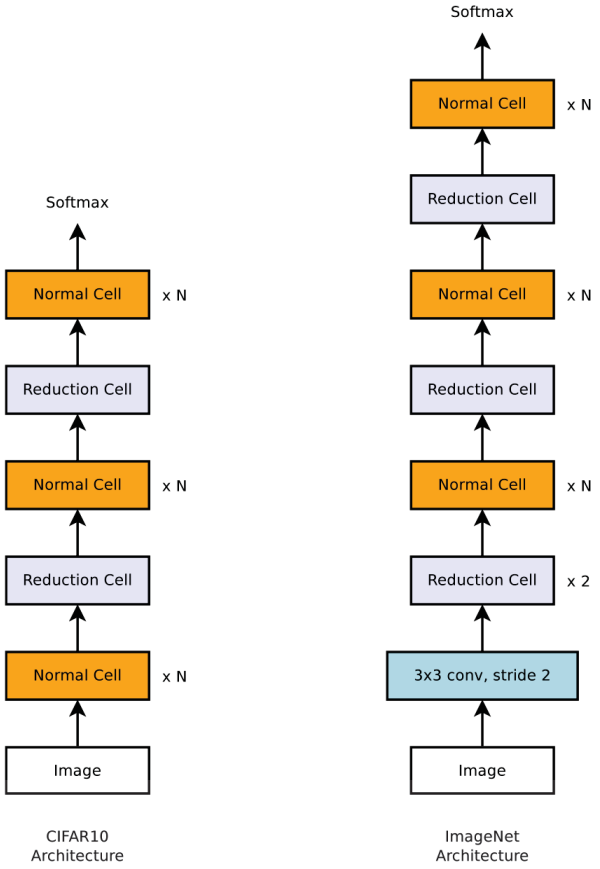


FIGURA 20 – NASNet. Diferentes combinações de células ”normais” e de ”redução” geradas para os datasets CIFAR-10 (à esquerda) e ImageNet (à direita). Fonte: Zoph, Vasudevan et al. (2018).

A acurácia Top-1 e Top-5 referem-se ao desempenho do modelo no conjunto de validação do ImageNet. O primeiro caso, acurácia Top-1, verifica se a classe inferida com maior probabilidade é a mesma da classe alvo, enquanto o segundo caso, Top-5, verifica se a classe alvo está entre as inferências com as 5 maiores probabilidades.

Por fim, a profundidade refere-se à profundidade topológica da rede, incluindo camadas de convolução, camadas de ativação, camadas de normalização em lote, e assim por diante.

TABELA 1 – COMPARATIVO ENTRE OS MODELOS DE CNNS

Modelo	Tamanho (MB)	Top-1 Acc	Top-5 Acc	Params	Camadas
DenseNet121	33	75,00%	92,30%	8,1M	242
InceptionV3	92	77,90%	93,70%	23,9M	189
NASNetMobile	23	74,40%	91,90%	5,3M	389
ResNet50V2	98	76,00%	93,00%	25,6M	103
VGG16	528	71,30%	90,10%	138,4M	16
VGG19	549	71,30%	90,00%	143,7M	19
Xception	88	79,00%	94,50%	22,9M	81

FONTE: Chollet et al. (2024)

2.2.5 Vision Transformer

Arquiteturas baseadas no mecanismo de *self-attention*, em particular as que utilizam Transformers (VASWANI et al., 2017), têm-se consolidado como a abordagem preferida na área de processamento de linguagem natural (do inglês, Natural Language Processing – NLP). Já na área da visão computacional, arquiteturas convolucionais ainda possuem grande popularidade, embora a presença das arquiteturas que fazem uso dos Transformers esteja se tornando cada vez mais comum. Um dos motivos para isto estar acontecendo diz respeito à criação do Vision Transformer - ViT, proposto por Dosovitskiy et al. (2021), os quais tendo em vista o sucesso e escalabilidade dos Transformers em NLP realizaram a aplicação do Transformer padrão, com o mínimo de modificações possíveis, para o contexto de imagens. A FIGURA 21 apresenta um panorama geral do Vision Transformer, ressaltando, à direita, a estrutura do Transformer.

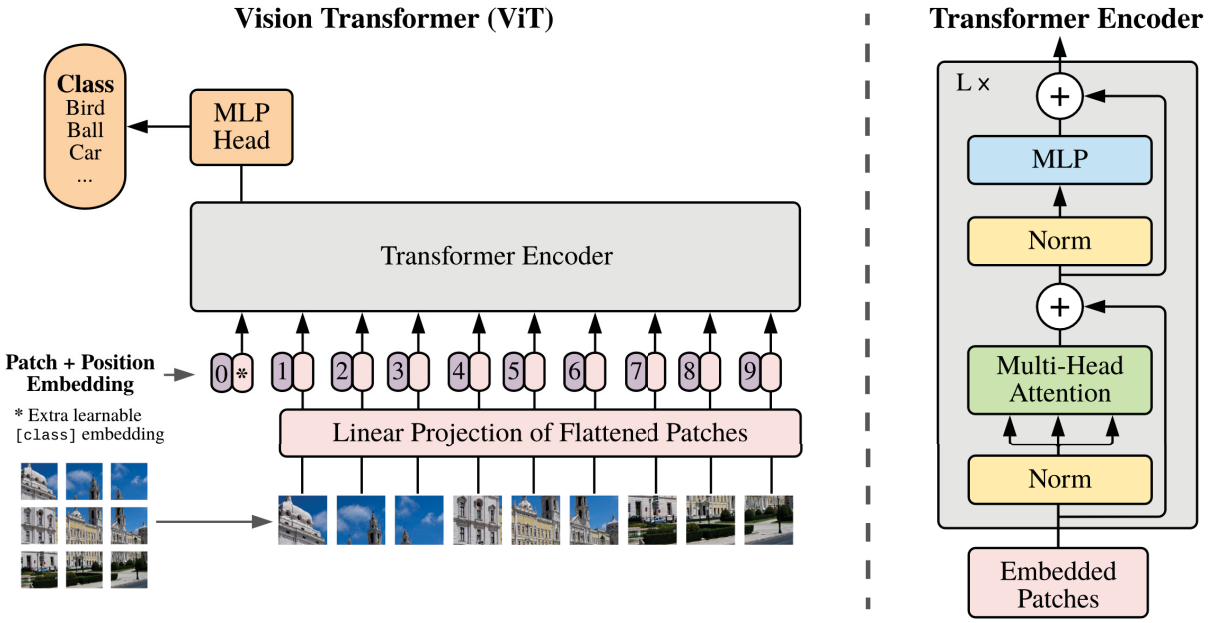


FIGURA 21 – Panorama geral do Vision Transformer. A imagem é dividida em *patches* de tamanho fixo, aplica-se uma codificação linear a cada um deles, e são adicionadas representações posicionais, fornecendo uma sequência resultante de vetores a um codificador Transformer padrão. Por fim, é adicionado um MLP para realizar a classificação. Fonte: Dosovitskiy et al. (2021)

Na estrutura do ViT, uma imagem é dividida em segmentos (*patches*) e a sequência de representações vetoriais (*embeddings*) lineares desses patches é fornecida como entrada para um Transformer, tratando-se os patches da mesma maneira que se faz com as palavras (*tokens*) em uma aplicação de NLP. Uma vez que o Transformer padrão recebe como entrada uma sequência unidimensional de embeddings de tokens, para este ser capaz de lidar com imagens bidimensionais a imagem é primeiramente transformada em uma sequência de patches 2D ”achatados”. Na sequência, ”embeddings posicionais” são acrescentados aos embeddings dos patches para preservar a informação posicional. A sequência resultante de vetores de embedding é, então, inserida no codificador (*transformer encoder*).

O codificador Transformer (FIGURA 21, à direita) é composto por camadas alternadas

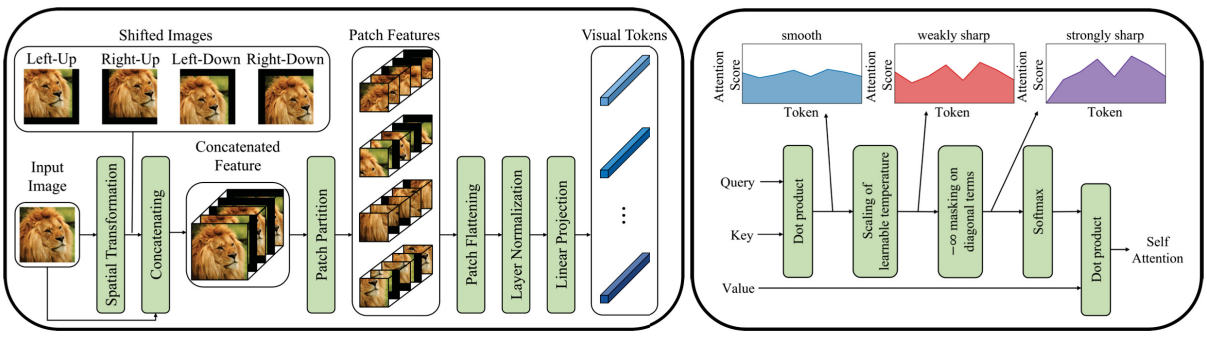
de ”atenção multi-head” e blocos MLP. Além disso, antes de cada bloco e conexão residual, aplica-se uma camada de normalização.

Segundo os autores, o Vision Transformer apresenta um viés indutivo específico de imagem menor em comparação com as CNNs. Nas CNNs, localidade, estrutura de vizinhança bidimensional e equivariância translacional são incorporadas em cada camada ao longo do modelo. Já no ViT, apenas as camadas MLP são locais e translacionalmente equivariantes, enquanto as camadas de *self-attention* atuam de forma global.

Embora modelos baseados em ViTs tenham atingido o estado-da-arte em várias tarefas de visão computacional, estes resultados decorrem do pré-treinamento destes modelos com grandes conjuntos de dados, como o JFT-300M (SUN et al., 2017) – o qual possui 300 milhões de imagens.

Esta dependência de conjuntos de dados de grande escala é vista como o resultado da sua característica de possuir baixo viés indutivo de localidade. Sendo assim, Lee, Lee e Song (2021) identificaram dois problemas responsáveis por reduzir o viés indutivo de localidade, os quais limitam o desempenho do ViT. O primeiro deles é a tokenização insuficiente. O ViT divide uma imagem em patches de tamanhos iguais, não sobrepostos, projetando linearmente cada patch em um token visual. O segundo problema está no mecanismo de atenção. A dimensão de características dos dados de imagem é consideravelmente maior do que em NLP, de modo que o número de tokens embutidos torna-se inevitavelmente elevado. Com isto, o ViT não consegue focar localmente nos tokens visuais mais relevantes.

Tendo em vista as questões discutidas anteriormente, Lee, Lee e Song (2021) propuseram duas soluções para melhorar o viés indutivo de localidade do ViT, facilitando o modelo aprender a partir de conjuntos de dados de menor escala. Primeiramente, propõe-se a Tokenização de Patches Deslocados (do inglês, Shifted Patch Tokenization - SPT), apresentado na FIGURA 22 (a), para aproveitar melhor as relações espaciais entre pixels vizinhos no processo de tokenização. Em segundo lugar, propõe-se o mecanismo de Self-Attention Localizada (do inglês, Locality Self-Attention - LSA), apresentado na FIGURA 22 (b), o qual permite ao ViT focar melhor localmente.



(a) Shifted Patch Tokenization (b) Locality Self-Attention

FIGURA 22 – ViT para datasets reduzidos. Diagramas dos métodos de Shifted Patch Tokenization (à esquerda) e Locality Self-Attention Mechanism (à direita). Fonte: Lee, Lee e Song (2021).

Shifted Patch Tokenization – Conforme exemplificado na FIGURA 22 (a), a imagem de entrada passa por uma transformação na qual são geradas imagens deslocadas espacialmente em meia unidade do tamanho do patch e em quatro direções diagonais: superior-esquerda, superior-direita, inferior-esquerda e inferior-direita. Em seguida, estas imagens deslocadas são recortadas para manter o mesmo tamanho da imagem de entrada e, então, concatenadas com a entrada original. Este resultado então é dividido em patches não sobrepostos, os quais são achatados (*flattened*). Em seguida, os tokens visuais são obtidos por meio da aplicação de "normalização de camadas"(do inglês, Layer Normalization - LN) e de projeção linear. Como resultado deste processo, o SPT é capaz de embutir mais informações espaciais nos tokens visuais, aumentando o viés indutivo de localidade nos ViTs.

Locality Self-Attention Mechanism – Os mecanismos de LSA, exemplificados na FIGURA 22 (b), condicionam a atenção a patches próximos ou vizinhos em vez de distribuir uniformemente a atenção a todos os patches da imagem. Isto auxilia o modelo a aprender e enfatizar padrões locais, fazendo com que ele torne-se mais eficiente na compreensão de imagens, especialmente nos casos de conjuntos de dados menores. As duas técnicas centrais do LSA são a máscara diagonal e o escalonamento de temperatura ajustável (*learnable temperature scaling*). A máscara diagonal é responsável por atribuir valores (*scores*) maiores às relações inter-tokens através da exclusão de relações self-token da operação de softmax, fazendo com que a atenção do ViT se concentre mais nos outros tokens, em vez de focar nos seus próprios tokens. A segunda técnica do LSA é o escalonamento de temperatura ajustável, que permite que o ViT defina a temperatura do softmax de forma autônoma durante o processo de aprendizado. Em termos gerais, uma baixa temperatura no softmax intensifica a distribuição de pontuações, resolvendo o problema de suavização na distribuição dos scores de atenção.

2.2.6 Aprendizado supervisionado

Dá-se o nome de aprendizado supervisionado quando um conjunto de dados previamente rotulados é utilizado para treinar uma rede neural artificial ou outro classificador. Este conjunto é composto por informações de entrada, como imagens ou vetores de características presentes nestas imagens, por exemplo. Estas informações são associadas à um valor de saída, o qual é utilizado para classificar, detectar ou delinear algum objeto de interesse. O que distingue o aprendizado de máquina clássico do aprendizado profundo é que neste primeiro tipo o vetor das características dos dados de entrada geralmente é definido pelo desenvolvedor, enquanto no segundo, o processo de seleção e extração destas características é feito automaticamente pela rede neural artificial durante o processo de aprendizado.

No campo da visão computacional existem três tarefas mais populares, sendo elas classificação, detecção e segmentação semântica. No caso da classificação de objetos, relaciona-se uma determinada classe ou conjunto de classes ao dado de entrada, enquanto na detecção e segmentação de objetos é necessário destacar a área ou região onde os objetos de interesse se encontram. Isto tem como objetivo direcionar o aprendizado da rede neural artificial para estas regiões.

No processo de aprendizado e avaliação de uma rede neural artificial frequentemente são utilizadas três partições de dados, sendo elas denominadas partições de treino, validação e teste. Sendo a partição de treino utilizada de fato para o aprendizado dos pesos dos neurônios, este conjunto costuma ser maior do que os demais. Enquanto isso a partição de validação, sendo representativa do conjunto de teste, é utilizada para possibilitar uma melhor compreensão do processo de treinamento. Por fim, a partição de teste é a porção dos dados nunca antes vista pela rede, portanto, sendo utilizada para avaliar o modelo gerado.

Existem diversas maneiras de particionar os dados, sendo comum a utilização de 60% para treino, 20% para validação e 20% para teste, embora em casos de bancos de dados com poucos exemplos é comum encontrar o particionamento de 80% para treino e 20% para teste.

A FIGURA 23 mostra as duas curvas de aprendizado durante o treinamento de uma rede neural artificial. Nela é possível observar como a rede está aprendendo ao longo de suas iterações, ou seja, sua taxa de erro (em azul) está sendo reduzida. Destaca-se como a curva de validação (em laranja) se aproxima da curva do erro, até o momento em que elas começam a divergir. Quando isto ocorre, a rede passa a sofrer de *overfitting*, quando ao invés de aprender a generalizar, ela possivelmente começou a decorar dados do conjunto de treino. O momento em que os erros de treino e validação estiverem mais próximos e com valores mais baixos, é o ponto em que se deve parar o processo de treinamento, ou se escolher os pesos para tal iteração.

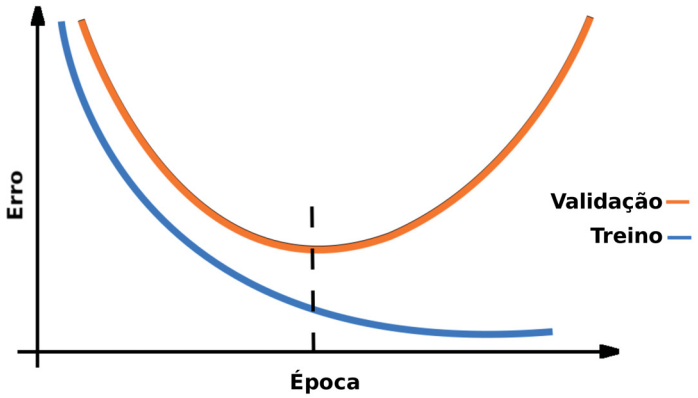


FIGURA 23 – Curvas de treino e validação. Em azul: curva de treino. Em laranja: curva de validação. Tracejado: ponto de parada ideal do treinamento. Fonte: Adaptado de Paweł Grabiński (2018).

Uma das melhores formas de estimar a performance de um modelo é através da técnica de validação cruzada, conhecida como k-fold (OZDEMIR, 2017). Este método de validação consiste na divisão do conjunto de dados em k partes iguais, sendo este valor geralmente igual ou superior a 5. Cada partição de dados é um *fold* (dobra, em português).

Neste método, a rede é treinada a mesma quantidade de vezes que a quantidade de dobras nas quais o conjunto de dados fora dividido. A cada treinamento, uma dobra distinta será retirada do total para ser utilizada como teste, deixando o restante das dobras para serem utilizadas no treinamento. Ao final de cada treinamento serão geradas as métricas de avaliação

(item 2.2.8) com o respectivo conjunto de validação. Por fim, tem-se que as métricas finais do modelo é a média das métricas obtidas para cada um dos treinamentos.

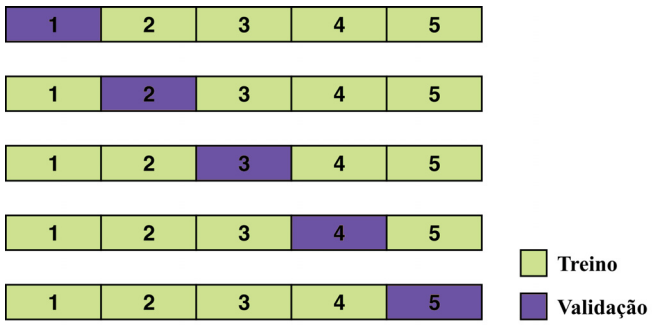


FIGURA 24 – Validação cruzada K-Fold. Validação cruzada com 5 dobras. Fonte: Adaptado de Restani (2019).

Esta técnica tem como vantagem permitir averiguar se o conjunto de dados tem uma distribuição de classes consistente, embora como desvantagem demande a necessidade de realizar diversos treinamentos. No caso de conjuntos de dados muito grandes e modelos de aprendizado profundo com muitos parâmetros isto pode se tornar um impeditivo, dado o custo computacional resultante.

2.2.7 Aumento de Dados

Uma técnica que permite extrapolar o conjunto original de dados e criar uma variabilidade maior de exemplos é a de aumento de dados (do inglês, data augmentation - DA). Sua implementação frequentemente resulta em uma melhoria no modelo da rede neural artificial, em função de colaborar com sua capacidade de melhor generalizar o problema.

O processo de aumento de dados, no caso de imagens, envolve a aplicação de diversas operações, aplicadas de forma individual ou em combinação, que alteram a imagem original. Essas modificações incluem cortes, rotações, inversões horizontais e verticais, ampliações, distorções, adição de ruído, entre outras. Isso resulta na criação de um novo conjunto de imagens derivadas a partir das originais, as quais são subsequentemente incorporadas ao banco de dados original.

A FIGURA 23 exemplifica este processo. Nela é possível observar uma imagem original, à esquerda, e as demais imagens dela resultantes, após a aplicação de diferentes operações de processamento digital de imagens.

2.2.8 Métricas de avaliação do aprendizado

Não há uma só métrica capaz de capturar todas as propriedades de um modelo de aprendizado de máquina. Sendo assim, geralmente várias métricas são apresentadas em conjunto para sintetizar a performance de um modelo (HICKS et al., 2022). Neste contexto, neste

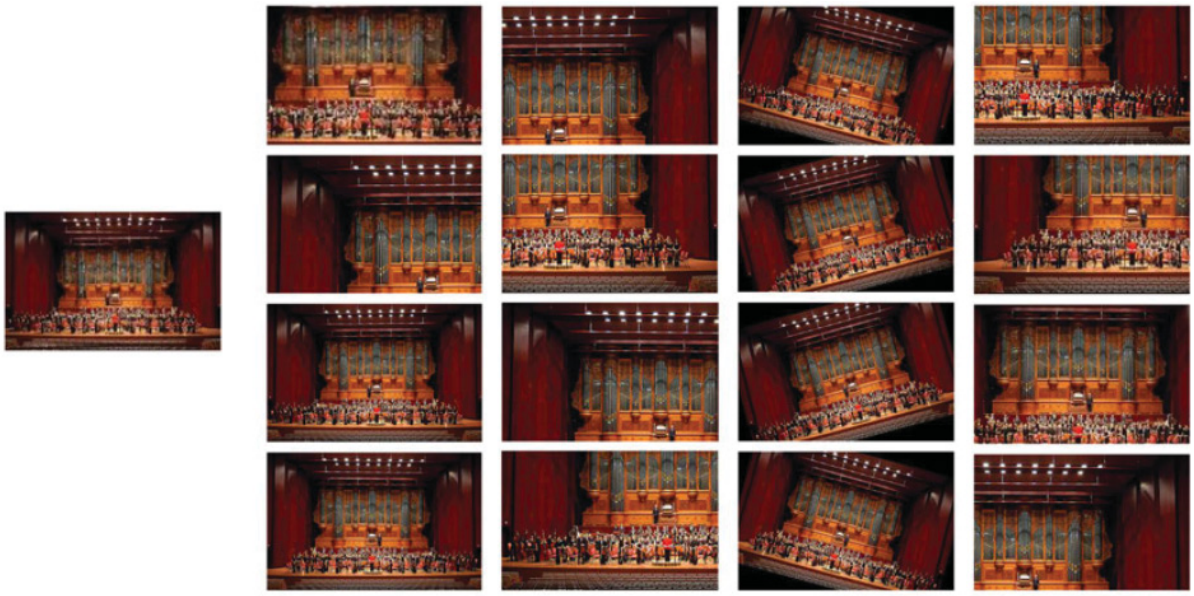


FIGURA 25 – Aumento de dados. À esquerda: imagem original. Demais imagens: resultado da técnica de aumento de dados. Fonte: Adaptado de Khan et al. (2018).

item será apresentado um conjunto de métricas que permitem, de maneira geral, conhecer melhor as características da performance de um modelo.

Matriz de confusão – A matriz de confusão fornece uma visão do desempenho de um algoritmo supervisionado ao apresentar as frequências de classificação. Para criar a matriz de confusão, um conjunto de dados de teste é classificado manualmente e, em seguida, submetido à rede neural. Os resultados previstos pela rede são então comparados com os valores considerados corretos.

A FIGURA 26 mostra como os resultados são dispostos. Os falsos positivos (FP) são as classificações nas quais o algoritmo identificou o dado como sendo de um determinada classe de maneira equivocada. Os verdadeiros positivos (VP), são as classificações nas quais o algoritmo fez a predição do dado de maneira correta. Os verdadeiros negativos (VN), são as classificações nas quais o algoritmo corretamente identificou que um dado não pertencia àquela classe. E, por fim, os falsos negativos (FN), são as classificações nas quais o algoritmo classificou o dado como não pertencente à uma determinada classe, da qual ele de fato fazia parte, de maneira equivocada.

Acurácia – A acurácia (*accuracy*, em inglês) é definida pela soma dos verdadeiros positivos (VP) com os verdadeiros negativos (VN), dividido pelo total de elementos. Desta forma, a acurácia mede o quanto o algoritmo acertou do total de predições realizadas.

$$\text{acurácia} = \frac{\text{VP} + \text{VN}}{\text{Total}} \tag{2.11}$$

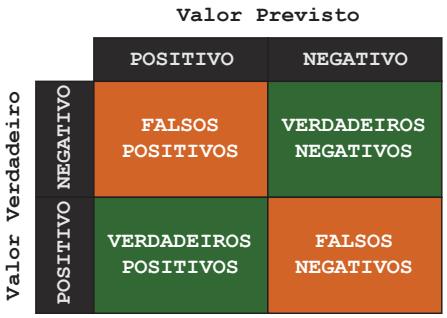


FIGURA 26 – Matriz de confusão. Matriz com valores previstos pelo modelo e os valores reais. Fonte: Restani (2019).

Precisão – A precisão (*precision*, em inglês) é uma métrica que permite avaliar quantas das predições dadas como positivas estão realmente corretas. Ela é dada pelo número de verdadeiros positivos (VP), dividido pela soma de verdadeiros positivos com os falsos positivos.

$$\text{precisão} = \frac{\text{VP}}{\text{VP} + \text{FP}} \tag{2.12}$$

Desta forma, a precisão mede para as classificações dadas como positivas a proporção que foi classificada de maneira correta entre todos os classificados como positivos. Um classificador com alta precisão será então um classificador que gera poucos falsos positivos.

Sensibilidade – A sensibilidade (ou *sensitivity*, do original em inglês) é uma métrica que permite avaliar quantos dos verdadeiros positivos foram identificados de maneira correta. Ou seja, ela é a taxa de verdadeiros positivos. A sensibilidade é dada pelo número de verdadeiros positivos (VP), dividido pela soma de verdadeiros positivos (VP) com falsos negativos (FN).

$$\text{sensibilidade} = \frac{\text{VP}}{\text{VP} + \text{FN}} \tag{2.13}$$

Desta forma, a sensibilidade mede para as previsões positivas a proporção que é classificada como negativa de maneira equivocada. Um classificador com alta sensibilidade será então um classificador que gera poucos falsos negativos.

Especificidade – A especificidade, ao contrário da sensibilidade, é a taxa de verdadeiros negativos. É dada pela quantidade de verdadeiros negativos (VN) dividido pela soma dos verdadeiros negativos (VN) e dos falsos positivos (FP).

$$\text{especificidade} = \frac{\text{VN}}{\text{VN} + \text{FP}} \tag{2.14}$$

Sendo assim, um classificador com alta especificidade será então um classificador que gera poucos falsos positivos.

F-score – A métrica F-score é dada pela média harmônica entre precisão e sensibilidade, congregando em um só valor estas duas métricas. Sendo assim, ela aplicará uma penalização caso o valor de alguma das métricas que a compõem seja extremo. A F-score é dada por 2 vezes a precisão multiplicada pela sensibilidade, dividida pela soma da precisão pela sensibilidade.

$$\text{F-score} = 2 \times \frac{precisao \times sensibilidade}{precisao + sensibilidade} \tag{2.15}$$

Curva ROC e AUC – A acurácia, por si só, não é uma boa medida para problemas em que há desbalanceamento de classes. Isto é, quando existem mais exemplos de uma determinada classe em detrimento de outra. Assim sendo, uma acurácia alta pode não refletir como o algoritmo é capaz de corretamente classificar a classe minoritária de um problema.

As limitações da “acurácia” quanto medida de um teste diagnóstico fazem necessários os conceitos de sensibilidade e especificidade. Estas métricas representam de certa forma dois tipos de acurácia, sendo elas os casos verdadeiramente positivos e os casos verdadeiramente negativos, respectivamente. Contudo, estas métricas dependem da seleção de um limiar (threshold, em inglês) de decisão arbitrário, fazendo com que elas não forneçam uma descrição única do desempenho diagnóstico. Uma maneira de visualizar os efeitos da variação deste limiar nos valores da sensibilidade e da especificidade é a curva de Característica de Operação do Receptor (em inglês, Receiver Operating Characteristic – ROC) (METZ, 1978).

A curva ROC é construída ao plotar no eixo y a taxa de verdadeiros positivos (sensibilidade) e no eixo x a taxa de falsos positivos (1 - especificidade), variando o limiar de decisão entre 0 e 1.

A FIGURA 27 exemplifica as características de curvas ROC, na qual são apresentados em vermelho, um classificador típico; em azul, um classificador perfeito; e através de uma linha tracejada, um classificador aleatório.

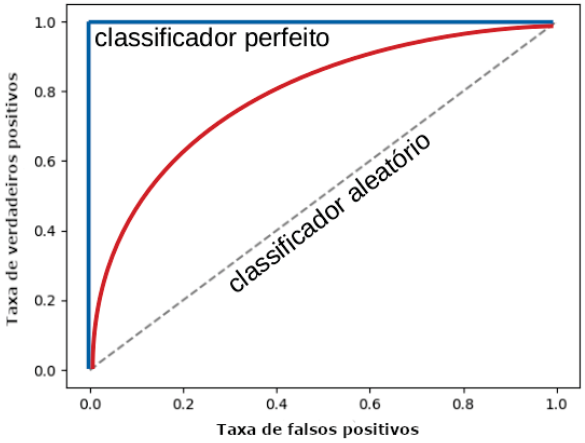


FIGURA 27 – Curva ROC. Características da curva ROC. Fonte: O autor (2023).

Embora a curva ROC forneça uma visualização detalhada do comportamento do classificador, ainda é difícil comparar de maneira rápida várias curvas. Uma maneira de condensar a informação de uma curva ROC em um número é calculando a área sob a curva (em inglês, Area Under the Curve - AUC). A FIGURA 28 exemplifica a AUC, à esquerda, do classificador perfeito, no qual este valor é igual a 1, e à direita, um classificador aleatório, no qual este valor é 0,5.

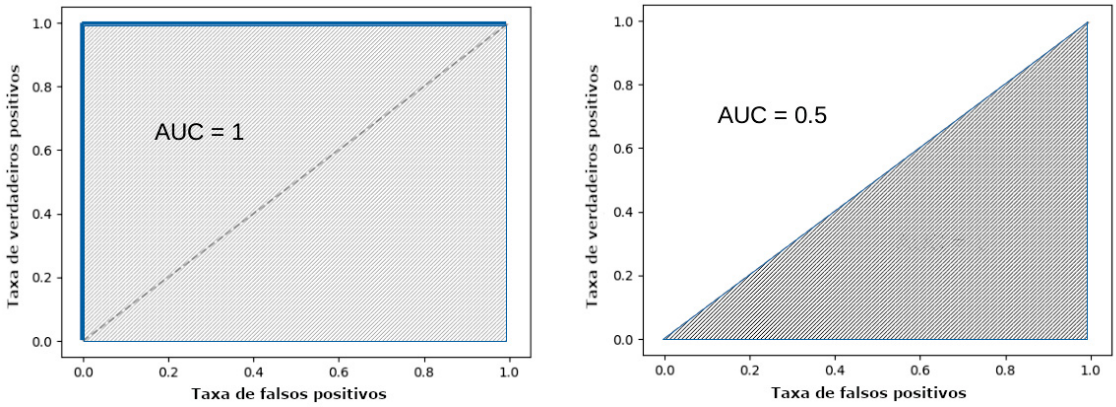


FIGURA 28 – Métrica AUC. Área sob a curva ROC, que resulta na métrica AUC. Fonte: O autor (2023).

3 REVISÃO DE LITERATURA

Neste capítulo são apresentados os trabalhos que se assemelham ao desenvolvimento proposto, bem como aqueles que tem relação mais direta com a detecção do aumento do átrio esquerdo ou do coração em radiografias.

Para realizar a detecção assistida por computador na distinção entre resultados radiográficos normais e anormais em imagens de tórax de cães, Yoon, Hwang e Lee (2018) avaliaram a viabilidade do uso das técnicas de *bag-of-features* (BOF) e redes neurais convolucionais (CNN). Para isto, foi construído um dataset com radiografias torácicas computadorizadas contendo projeção ventrodorsal e lateral, na mesma escala, com as seguintes categorias: (1) silhueta cardíaca normal vs. cardiomegalia, (2) pulmão normal vs. padrões pulmonares anormais, (3) posição mediastinal normal vs. deslocamento mediastinal, (4) espaço pleural normal vs. derrame pleural e (5) espaço pleural normal vs. pneumotórax. Como resultado, a CNN mostrou maior precisão (CNN; entre 92,9 e 96,9% e BOF; 79,6 e 96,9%) e sensibilidade (CNN; entre 92,1 e 100% e BOF; entre 74,1 e 94,8%) do que o BOF. Além disso, a CNN obteve maior acurácia do que o BOF para todos os achados radiográficos.

Para realizar a classificação de radiografias torácicas de cães, as quais foram anotadas com multiplos rótulos, Banzato, Wodzinski, Burti et al. (2021) também desenvolveram um método baseado em rede neural convolucional (CNN). Os achados radiográficos usados como rótulos para treinar as CNNs foram: normal, cardiomegalia, padrão alveolar, padrão brônquico, padrão intersticial, massa, derrame pleural, pneumotórax e megaesôfago. Foram desenvolvidas e testadas duas CNNs diferentes, com base nas arquiteturas ResNet-50 e DenseNet-121. A CNN baseada em ResNet-50 apresentou uma AUC acima de 0,8 para todos os achados radiográficos incluídos, exceto para padrões brônquicos e intersticiais. Já a CNN baseada em DenseNet-121 teve um desempenho geral inferior.

Outro trabalho relacionado é o de Nam et al. (2021), os quais implementaram algoritmos de aprendizado profundo para realizar a detecção de aumento das câmaras cardíacas para o átrio esquerdo (baseado na rede DenseNet) e para o ventrículo (baseado na rede ResNet152), porém em radiografias torácicas de humanos. Foram coletadas 5.045 radiografias, além de 107 para uma validação. Um teste de desempenho foi conduzido, no qual cinco radiologistas cardio-torácicos avaliaram as radiografias sem e com os resultados dos modelos. O algoritmo superou o desempenho dos radiologistas cardio-torácicos na detecção de aumento do átrio esquerdo e mostrou promessa na triagem de indivíduos com aumento moderado a grave do átrio esquerdo em um programa de triagem de saúde. Além disso, os radiologistas cardio-torácicos melhoraram seu desempenho na detecção do aumento do átrio esquerdo quando auxiliados pelo algoritmo.

Também é possível encontrar trabalhos que avaliaram o desempenho das CNNs na realização de medição da Vertebral Heart Scale (BOISSADY et al., 2021). Os autores utilizaram

radiografias de cães e gatos, e realizaram a comparação dos resultados com dois especialistas certificados. A técnica implementada foi baseada na arquitetura DenseNet-121. 30 radiografias laterais torácicas caninas e 30 felinas foram avaliadas por cada operador, usando dois métodos diferentes para determinação do eixo curto cardíaco nas radiografias de cães: a abordagem original publicada por Buchanan e Bücheler (1995) e a abordagem modificada proposta pelos autores do ensaio EPIC. Apenas o método de Buchanan foi usado para as radiografias de gatos. No geral, o VHS calculado pela IA, pelos radiologistas e pelos cardiologistas apresentou alto grau de concordância tanto em pacientes caninos quanto felinos (coeficiente de correlação intraclasse (CCI) = 0,998).

Saxena, Sastry e Roopashree (2023) também desenvolveram uma abordagem para categorizar radiografias torácicas caninas, inclusive com a presença de cardiomegalia, usando Redes Neurais Profundas. Radiografias torácicas foram coletadas retrospectivamente de 2010 a 2020. Os dados das radiografias foram divididos ao meio, uma vez que provenientes de dois métodos distintos de aquisição de radiografias. A generalização das redes foi avaliada usando o Conjunto de Dados 2, enquanto o Conjunto de Dados 1 foi utilizado para treinamento e teste. Os rótulos utilizados foram: normal, padrão brônquico, cardiomegalia, massa, derrame pleural, padrão alveolar, padrão intersticial, pneumotórax e megaesôfago. As redes desenvolvidas foram baseadas nas arquiteturas ResNet-50 e DenseNet-121. AUCs acima de 0,8 foram alcançados pela rede baseada em ResNet-50 para todos os achados radiográficos incluídos nos Conjuntos de Dados 1 e 2, exceto padrões brônquicos e intersticiais. O desempenho geral da rede baseada em DenseNet-121 foi inferior.

Mais diretamente com relação à doença da válvula mitral mixomatosa (MMVD), Valente et al. (2023) desenvolveram um algoritmo com base em inteligência artificial (IA) para classificar diferentes estágios da doença em radiografias torácicas de cães. Das radiografias selecionadas, apenas as que claramente mostravam a silhueta cardíaca foram consideradas, e então foram classificadas segundo as diretrizes do American College of Veterinary Internal Medicine (ACVIM). Os cães assintomáticos sem sinais de remodelação cardíaca radiográfica ou ecocardiográfica foram classificados como B1, se a cardiomegalia com aumento do átrio e ventrículo esquerdos fosse evidente como B2, animais com pelo menos um episódio de edema pulmonar e/ou derrame pleural devido à ICC foram considerados estágio C, e cães sintomáticos refratários ao tratamento cardíaco padrão foram classificados como estágio D. A rede ResNet18 foi treinada em visualizações laterais direitas e esquerdas e/ou ventro-dorsais ou dorso-ventrais. A área sob a curva (AUC) mostrou um bom desempenho na determinação do estágio da MMVD a partir das visualizações laterais, com um AUC de 0,87, 0,77 e 0,88 para os estágios B1, B2 e C + D, respectivamente.

Também tratando do escore cardíaco vertebral (VHS), Solomon et al. (2023) validaram o uso de um algoritmo deste escore em comparação com a pontuação manual realizada por três cardiologistas veterinários certificados. Foi desenvolvida uma CNN de segmentação semântica para prever o tamanho do coração e as vértebras. Essas previsões foram usadas para calcular o escore cardíaco vertebral. Três cardiologistas veterinários certificados pontuaram manualmente 400 imagens cada, usando o método tradicional de Buchanan. Após a pontuação,

os cardiologistas avaliaram o algoritmo quanto a pontos anatômicos mal alinhados e qualidade geral da imagem. A diferença absoluta no percentil 95 entre o escore cardíaco vertebral dos cardiologistas e o escore cardíaco vertebral do algoritmo foi de 1,05 vértebras (intervalo de confiança de 95%: 0,97 a 1,20 vértebras), com um viés médio de -0,09 vértebras (intervalo de confiança de 95%: -0,12 a -0,05 vértebras). Com isto, concluiram que o desempenho do algoritmo de escore cardíaco vertebral era comparável ao dos três cardiologistas.

3.1 TRABALHOS RELACIONADOS

Entre os trabalhos encontrados no levantamento realizado, dois se destacaram por serem diretamente relacionados ao desenvolvimento proposto.

O primeiro deles se trata de um projeto piloto desenvolvido com o objetivo de aplicar inteligência artificial em radiografias torácicas para a detecção de aumento do átrio esquerdo em cães e comparar os resultados obtidos com as interpretações de radiologistas veterinários (LI et al., 2020). Para isto, foram selecionadas 792 radiografias laterais direitas de pacientes caninos com radiografias torácicas para treinar, validar e testar um algoritmo de rede neural convolucional (CNN). Foi utilizada a técnica de validação cruzada e os resultados apresentados são relativos às médias dos modelos treinados. A precisão, sensibilidade e especificidade obtidas foram comparadas com as de radiologistas veterinários certificados por conselho, obtendo-se precisão de 82,71%, sensibilidade de 68,42% e especificidade 87,09%, usando uma variante do algoritmo de rede neural convolucional orientada para acurácia, e 79,01%, 73,68% e 80,64%, respectivamente, usando uma variante orientada para sensibilidade. Em comparação, a precisão, sensibilidade e especificidade alcançadas pelos radiologistas foram de 82,71%, 68,42% e 87,09%, respectivamente. Embora a acurácia geral do algoritmo de CNN orientado para acurácia e dos radiologistas veterinários tenha sido idêntica, a concordância entre as duas abordagens foi de 85,19%.

O segundo trabalho relacionado apresenta um fluxo diagnóstico, consistindo na coleta de dados, pré-processamento de dados, detecção e classificação de objetos e segmentação de imagens (OH et al., 2023). Para tal, foi utilizada a rede YOLOv5 para detectar o coração em imagens de raios-X e classificá-lo como normal ou anormal. Posteriormente, nos casos classificados como anormais, a segmentação de imagens demonstra visualmente o grau de aumento do átrio esquerdo. A acurácia de classificação atingiu 0,8800 para a classe normal e 0,8933 para a classe anormal, resultando em uma acurácia de classificação geral de 0,8866. Além disso, foi obtido um escore F1 de 0,8864 e um escore AUC de 0,8866. O desempenho de segmentação de imagens foi avaliado usando o escore de DICE, alcançando um desempenho médio de 0,9026.

4 METODOLOGIA

Este capítulo apresenta informações relacionadas ao desenvolvimento realizado, iniciando pela construção do *dataset*, com as configurações dos ambientes de trabalho e das arquiteturas de redes neurais artificiais na sequência.

4.1 DATASET

Para a criação do conjunto de dados foi realizado um estudo retrospectivo com o levantamento de imagens radiográficas torácicas do banco de imagens do Hospital Veterinário da Universidade Federal do Paraná (HV-UFPR), combinadas com uma seleção de imagens radiográficas torácicas de cães escolhidas por radiologistas brasileiros de outros cinco hospitais veterinários. Os critérios de inclusão foram imagens bem posicionadas e de boa qualidade diagnóstica, nas quais foi possível delinear toda a silhueta cardíaca.

Ao todo, para este dataset, foram selecionados 450 pacientes, somando um total de 1039 radiografias látero-laterais, com resoluções entre 4248 × 3480 pixels e 2328 × 1728 pixels, com 72 x 72 ppi e 8-bits de profundidade de escala cinza. Estas imagens, originalmente em formato DICOM, foram convertidas para PNG. Nos casos em que as imagens foram geradas através do escaneamento da radiografia, era possível notar em alguns arquivos a presença de bordas pretas, como pode ser visto na FIGURA 29-A. Sendo assim, estas imagens tiveram as áreas com informação da radiografia (parte interna do tracejado vermelho na FIGURA 29-B) recortadas para gerar a imagem final (FIGURA 29-C). Por fim, as imagens foram redimensionadas para uma largura de 640 pixels, enquanto a proporção original da imagem foi mantida.

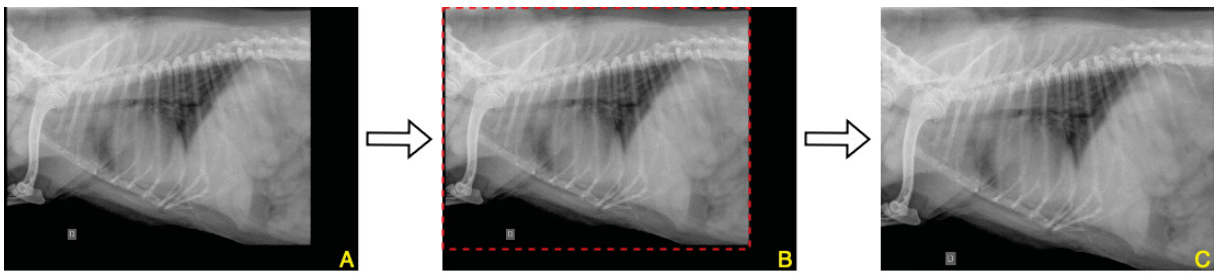


FIGURA 29 – Processo de recorte das imagens geradas através de escaneamento das radiografias. (A) imagem original. (B) área com informação tracejada em vermelho, (C) imagem resultante. Fonte: O autor (2024).

O casos foram classificados pela médica veterinária Lorena Tavares de Brito Nery Jaworsli, doutoranda em Ciências Veterinárias com 5 anos de experiência em diagnóstico por imagem, e revisados pela Profª Dra. Tilde Rodrigues Froes, ambas do Hospital Veterinário da Universidade Federal do Paraná - UFPR.

O dataset é composto por três grupos de imagens de radiografias torácicas de cães, sendo estes 98 pacientes que apresentam aumento do átrio esquerdo (AAE), FIGURA 31, 62 pacientes que apresentam aumento do átrio esquerdo e edema pulmonar (AAE_EP), FIGURA 32, e 290 pacientes considerados normais (N), FIGURA 30, resultando nos 450 pacientes.

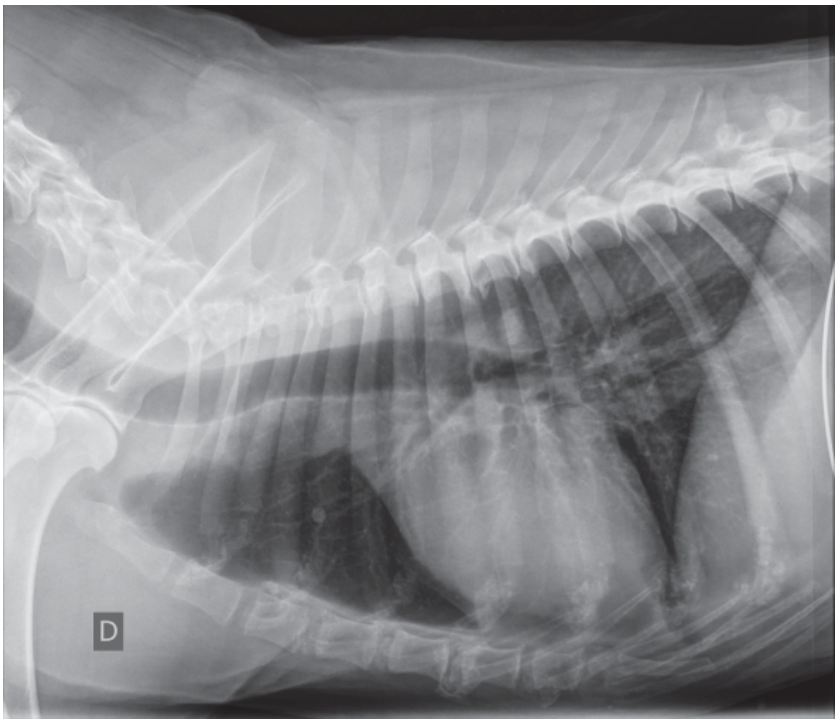


FIGURA 30 – Paciente Normal. Radiografia torácica de paciente normal. Fonte: O autor (2023).

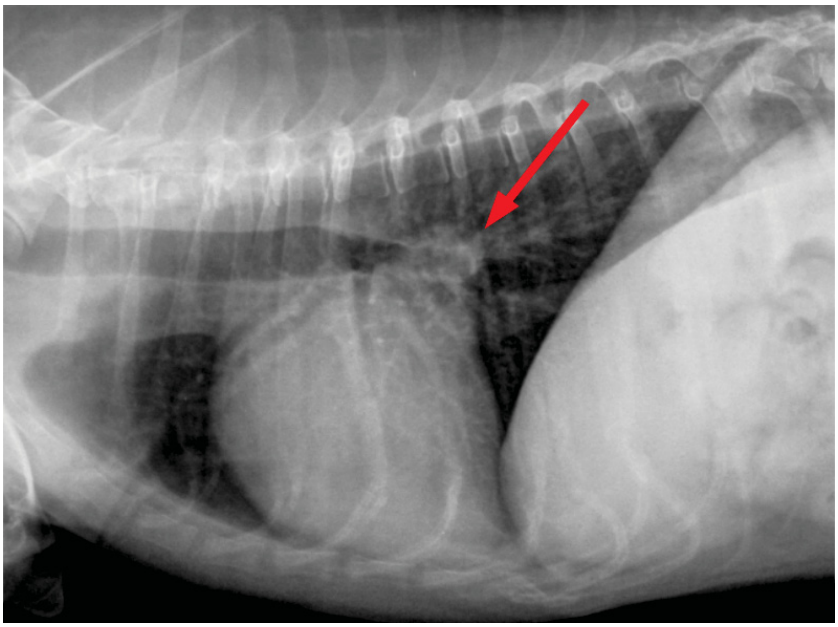


FIGURA 31 – Paciente com Aumento do átrio esquerdo (AAE). Radiografia torácica com seta vermelha apontando para o átrio esquerdo aumentado. Fonte: O autor (2023).

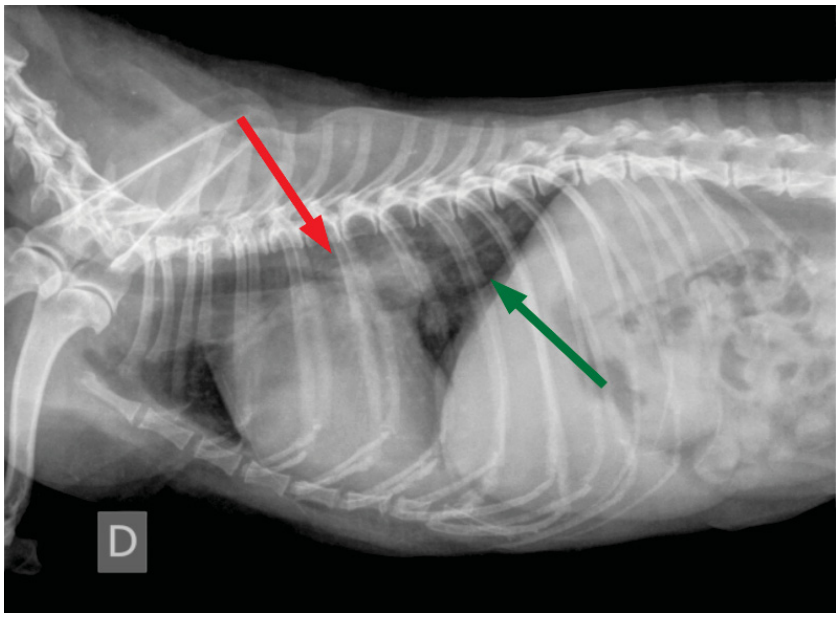


FIGURA 32 – Paciente com Aumento do átrio esquerdo e edema pulmonar (AAE_EP). Radiografia torácica com seta vermelha apontando o átrio esquerdo aumentado e seta verde apontando a presença de edema pulmonar. Fonte: O autor (2023).

Uma vez que o objetivo deste experimento foi detectar apenas as radiografias nas quais o paciente apresente o aumento do átrio esquerdo, os conjuntos AAE e AAE_EP foram agrupados, somando um total de 160 pacientes.

Por fim, os pacientes foram manualmente separados em 5-folds, conforme a configuração apresentada na TABELA 2.

TABELA 2 – DISTRIBUIÇÃO DOS PACIENTES NOS 5-FOLDS

Fold	AAE	AAE_EP	AAE + AAE_EP	Imagens	N	Imagens
1	19	13	32	70	58	122
2	19	12	31	66	58	134
3	20	12	32	82	58	141
4	20	12	32	67	58	147
5	20	13	33	66	58	144
Total	98	62	160	351	290	688

FONTE: O autor (2023).

4.2 CONFIGURAÇÕES

Os experimentos foram conduzidos através do Google Colab, no qual foi alocada uma máquina com processador Intel(R) Xeon(R) CPU de 2.30GHz, memória RAM (Random-access memory) de 13 GB (Gigabytes), e uma GPU (Graphics processing unit) NVIDIA Tesla T4.

O sistema operacional utilizado foi o Linux distribuição Ubuntu 22.04.2 LTS, com biblioteca de computação paralela CUDA (Compute Unified Device Architecture) versão 11.8.

As redes neurais artificiais foram implementadas utilizando a linguagem de programação Python, versão 3.10.12 e as bibliotecas tensorflow e keras, ambas na versão 2.15.0. As métricas foram geradas através das bibliotecas scikit-learn, versão 1.2.2.

Foi adicionada à arquitetura das CNNs uma camada de GlobalAveragePooling2D, uma camada de 1024 neurônios com função de ativação 'reLu', e, por fim, um neurônio com função de ativação 'sigmoid' para realizar a classificação binária, conforme representado na FIGURA 33.

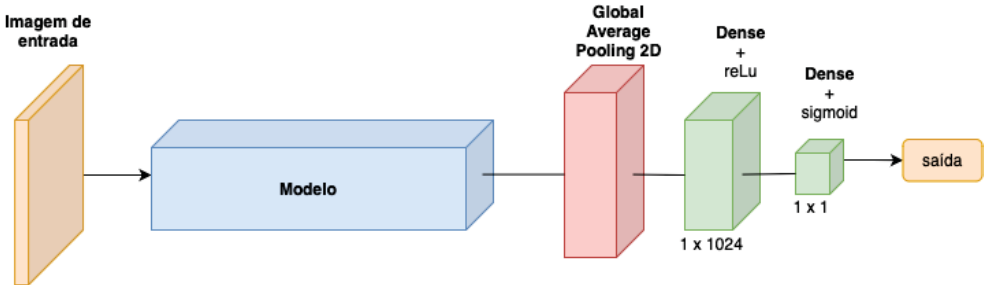


FIGURA 33 – Configuração das CNNs. É utilizada a estrutura principal de extração de características dos modelos (em azul), sem o tronco original de classificação. Esta estrutura é conectada a uma camada de Global Average Pooling 2D, em vermelho, uma camada totalmente conectada com função "reLu" e uma camada totalmente conectada com função "sigmoid", ambas em verde. Fonte: O autor (2024).

Foi utilizado como otimizador o algoritmo de descida gradiente estocástica (do inglês, *Stochastic Gradient Descent* - SGD). O SGD foi escolhido por ser eficiente e de simples implementação, permitindo um aprendizado rápido já nas etapas iniciais do treinamento. Para melhorar a estabilidade do SGD, foi utilizada uma taxa de aprendizado decrescente, a qual partiu inicialmente de 0,01, sendo reduzida pela metade a cada 10 épocas de treinamento. Estes valores foram escolhidos de maneira empírica.

Como o problema trata de uma classificação binária (classe "normal" ou "aumento do átrio esquerdo") e o modelo retorna uma probabilidade usando uma função de ativação sigmoide, a função objetiva escolhida foi a entropia cruzada binária (do inglês, *Binary Cross-Entropy* - BCE).

A dimensão de entrada das redes foi de 416 x 416, com um processamento em lote (do inglês, *batch*) de tamanho 4.

Foram realizados 5 treinamentos para cada rede sem aumento de dados e 5 treinamentos com aumento de dados. A cada treinamento, 4 folds foram utilizados para esta finalidade, e 1 utilizado como conjunto de validação. Ou seja, no primeiro treinamento o fold 01 foi retirado para validação, no segundo treinamento o fold 02 foi retirado para validação, e assim por diante.

As funções de aumento de dados utilizadas, tanto para as CNNs quanto para os ViTs foram: rotação de $\pm$ 20 graus; zoom de $\pm$ 5%, deslocamento vertical e deslocamento horizontal de $\pm$ 20%; e ruído gaussiano.

5 RESULTADOS E DISCUSSÃO

Nesta seção, são apresentados os resultados obtidos a partir da aplicação dos diferentes algoritmos selecionados em conjunto com o *dataset* confeccionado, além da discussão subsequente. A apresentação dos resultados está organizada em cinco subseções. A primeira congrega os resultados obtidos através da implementação das CNNs, a segunda congrega os resultados obtidos através da implementação de Transformers, a terceira apresenta as melhorias que podem ser obtidas realizando combinações entre os modelos, a quarta faz um compilado de todos os resultados obtidos, com o objetivo de facilitar sua interpretação, e a quinta – e última – apresenta resultados visuais das camadas de convolução ativadas durante as inferências através do algoritmo Grad-CAM (SELVARAJU et al., 2020).

O desempenho dos modelos foi avaliado utilizando as métricas de acurácia, sensibilidade, especificidade, precisão, F-score e AUC, as quais permitem analisar diferentes aspectos dos modelos obtidos.

5.1 MODELOS BASEADOS EM CNNS

No total, foram treinados 7 modelos diferentes de CNNs, sendo eles: InceptionV3, DenseNet-121, NASNet Mobile, ResNet50V2, VGG16, VGG19, e Xception. Todos eles foram treinados com e sem a assistência da técnica de aumento de dados. Estas arquiteturas de CNNs foram escolhidas por frequentemente aparecerem em trabalhos de relacionados à classificação de radiografias, seja na área de medicina veterinária (BURTI et al., 2020; GOMES et al., 2021; SAXENA; SASTRY; ROOPASHREE, 2023), de medicina humana (MUJAHID et al., 2022; FAN et al., 2023), ou até mesmo em outros tipos de imagem médica (SAJID et al., 2023).

A partir dos treinamentos foram obtidas as métricas de acurácia, precisão, sensibilidade, especificidade, F-score e AUC, para cada um dos 5 *folds*. Nos resultados apresentados em tabelas neste capítulo, as colunas nomeadas ”Fold” de 01 até 05 correspondem aos *folds* retirados para validação naquele treinamento. Além disso, também foram obtidas as médias dos resultados do treinamento de cada *fold* e seus respectivos desvios padrão.

A primeira CNN a ser treinada foi a InceptionV3. Os resultados obtidos podem ser observados na TABELA 3.

TABELA 3 – INCEPTIONV3

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7708	0,7100	0,7354	0,7056	0,8333	0,7510 ± 0,0472
Sensibilidade	0,7787	0,7090	0,6454	0,5986	0,7847	0,7033 ± 0,0730
Especificidade	0,7571	0,7121	0,8902	0,9403	0,9394	0,8478 ± 0,0953
Precisão	0,8482	0,8333	0,9100	0,9565	0,9658	0,9028 ± 0,0542
F-score	0,8120	0,7661	0,7552	0,7364	0,8659	0,7871 ± 0,0466
AUC	0,8738	0,7810	0,8767	0,8953	0,9475	0,8748 ± 0,0539

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 34.

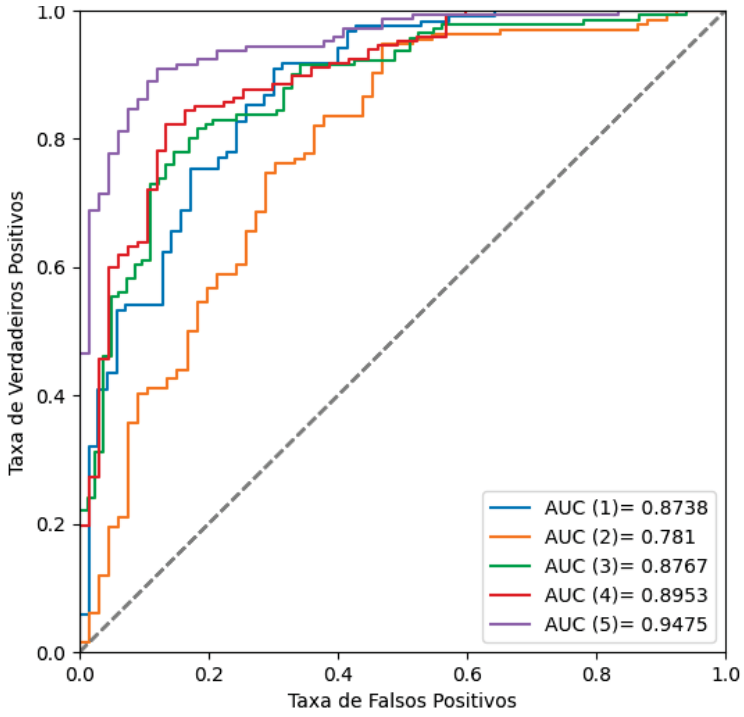


FIGURA 34 – Curvas ROC - InceptionV3. Curvas ROC e AUC obtidas para cada treinamento da InceptionV3. Fonte: O autor (2024).

A InceptionV3 também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 4.

TABELA 4 – INCEPTIONV3 - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7708	0,8100	0,7758	0,8832	0,8190	0,8118 $\pm$ 0,0403
Sensibilidade	0,6721	0,8731	0,6950	0,9320	0,8403	0,8025 $\pm$ 0,1017
Especificidade	0,9429	0,6818	0,9146	0,7761	0,7727	0,8176 $\pm$ 0,0972
Precisão	0,9535	0,8478	0,9333	0,9013	0,8897	0,9051 $\pm$ 0,0365
F-score	0,7885	0,8603	0,7967	0,9164	0,8643	0,8452 $\pm$ 0,0474
AUC	0,8938	0,8376	0,8973	0,9395	0,9199	0,8976 $\pm$ 0,0342

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 35.

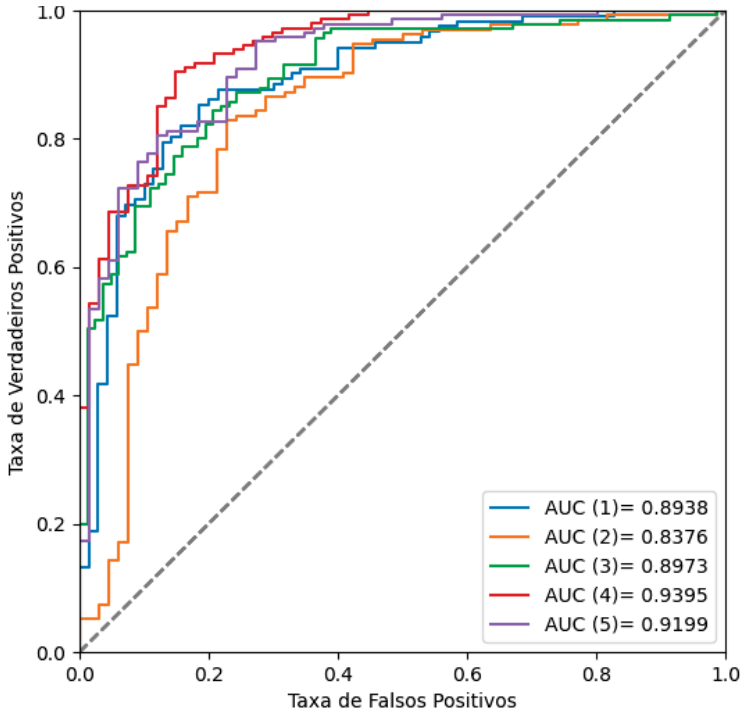


FIGURA 35 – Curvas ROC - InceptionV3 com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da InceptionV3. Fonte: O autor (2024).

A segunda CNN a ser treinada foi a DenseNet-121. Os resultados obtidos podem ser observados na TABELA 5.

TABELA 5 – DENSENET-121

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8073	0,7950	0,7848	0,8458	0,8381	0,8142 $\pm$ 0,0239
Sensibilidade	0,8197	0,9627	0,7376	0,8299	0,9097	0,8519 $\pm$ 0,0777
Especificidade	0,7857	0,4545	0,8659	0,8806	0,6818	0,7337 $\pm$ 0,1564
Precisão	0,8696	0,7818	0,9043	0,9385	0,8618	0,8712 $\pm$ 0,0523
F-score	0,8439	0,8629	0,8125	0,8809	0,8851	0,8571 $\pm$ 0,0266
AUC	0,8905	0,7834	0,9065	0,9401	0,9179	0,8877 $\pm$ 0,0546

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 36.

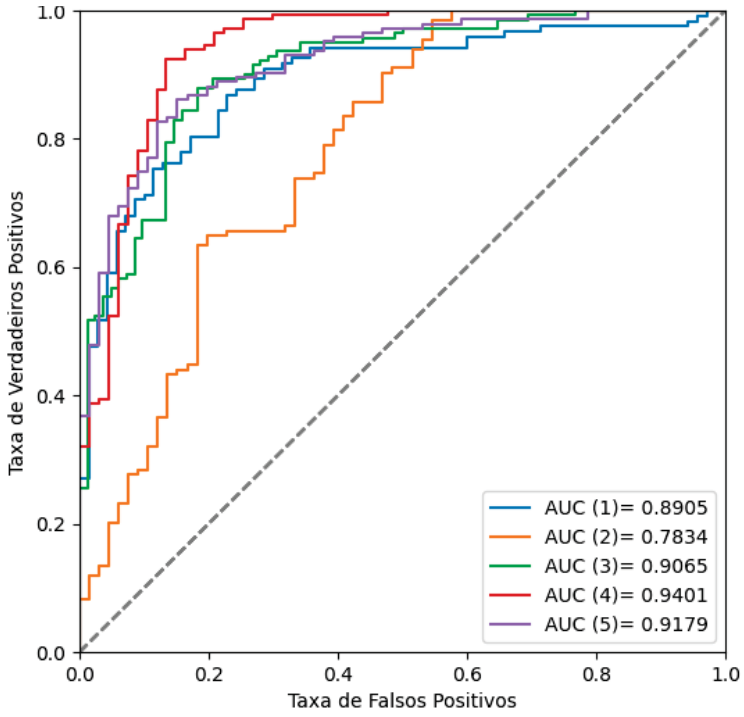


FIGURA 36 – Curvas ROC - DenseNet-121. Curvas ROC e AUC obtidas para cada treinamento da DenseNet-121. Fonte: O autor (2024).

A DenseNet-121 também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 6.

TABELA 6 – DENSENET-121 - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8281	0,7300	0,8341	0,8271	0,8476	0,8134 $\pm$ 0,0423
Sensibilidade	0,8607	0,7388	0,9362	0,7959	0,8611	0,8385 $\pm$ 0,0668
Especificidade	0,7714	0,7121	0,6585	0,8955	0,8182	0,7712 $\pm$ 0,0823
Precisão	0,8678	0,8390	0,8250	0,9435	0,9118	0,8774 $\pm$ 0,0444
F-score	0,8642	0,7857	0,8771	0,8635	0,8857	0,8552 $\pm$ 0,0357
AUC	0,8687	0,7965	0,9075	0,9454	0,9198	0,8876 $\pm$ 0,0518

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 37.

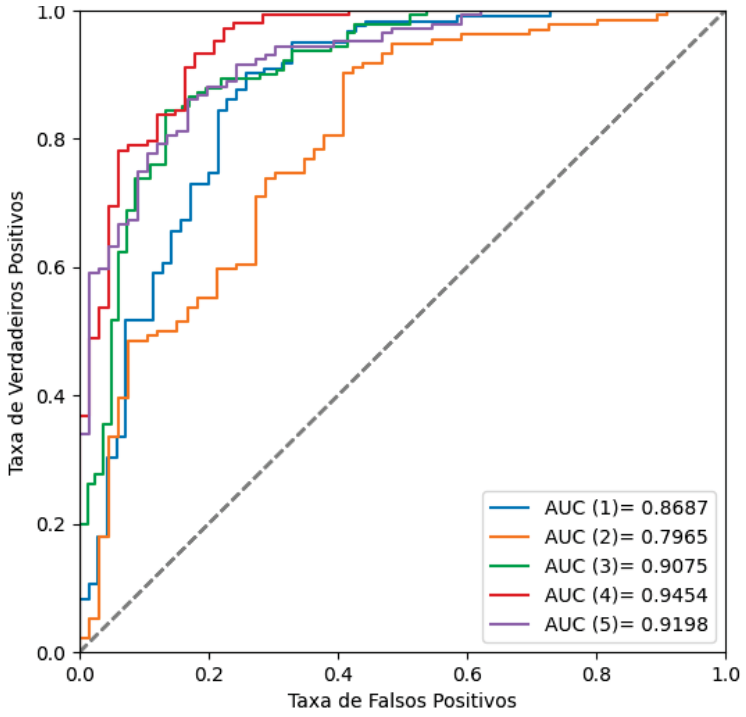


FIGURA 37 – Curvas ROC - DenseNet-121 com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da DenseNet-121. Fonte: O autor (2024).

A terceira CNN a ser treinada foi a NASNet Mobile. Os resultados obtidos podem ser observados na TABELA 7.

TABELA 7 – NASNET MOBILE

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,6875	0,6800	0,7668	0,8738	0,8238	0,7664 ± 0,0755
Sensibilidade	0,5328	0,7090	0,7163	0,9184	0,8472	0,7447 ± 0,1324
Especificidade	0,9571	0,6212	0,8537	0,7761	0,7727	0,7962 ± 0,1103
Precisão	0,9559	0,7917	0,8938	0,9000	0,8905	0,8864 ± 0,0530
F-score	0,6842	0,7480	0,7953	0,9091	0,8683	0,8010 ± 0,0809
AUC	0,8614	0,6807	0,8364	0,9050	0,8980	0,8363 ± 0,0817

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 38.

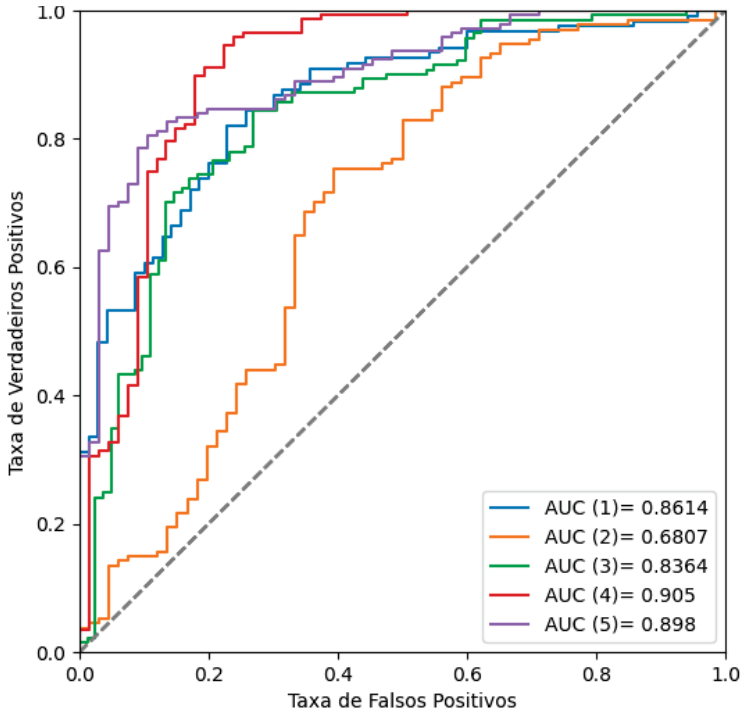


FIGURA 38 – Curvas ROC - NASNet Mobile. Curvas ROC e AUC obtidas para cada treinamento da NASNet Mobile. Fonte: O autor (2024).

A NASNet Mobile também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 8.

TABELA 8 – NASNET MOBILE - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8281	0,7800	0,8296	0,8505	0,8048	0,8186 $\pm$ 0,0241
Sensibilidade	0,9508	0,8582	0,8298	0,9116	0,9375	0,8976 $\pm$ 0,0464
Especificidade	0,6143	0,6212	0,8293	0,7164	0,5152	0,6593 $\pm$ 0,1062
Precisão	0,8112	0,8214	0,8931	0,8758	0,8084	0,8420 $\pm$ 0,0354
F-score	0,8755	0,8394	0,8603	0,8933	0,8682	0,8673 $\pm$ 0,0177
AUC	0,8658	0,7820	0,8906	0,9148	0,8392	0,8585 $\pm$ 0,0458

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 39.

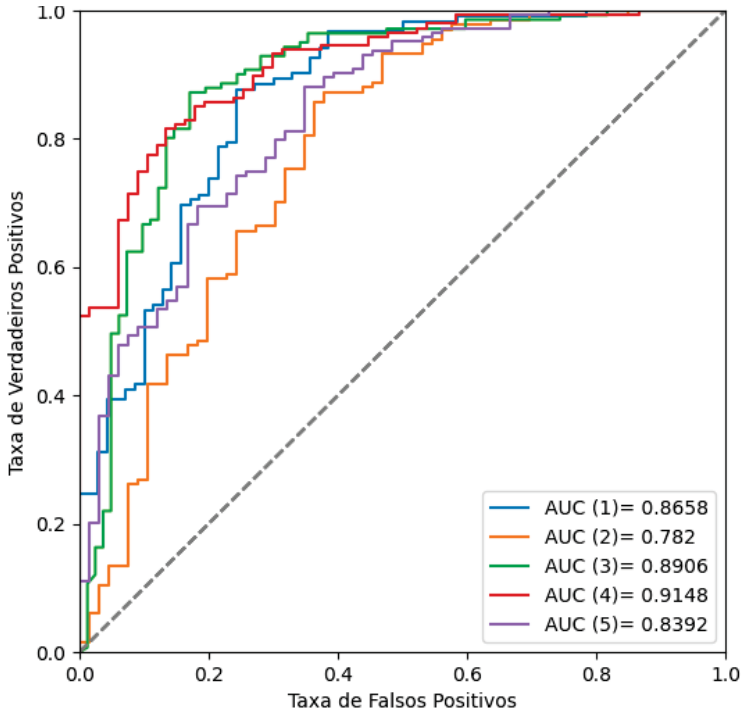


FIGURA 39 – Curvas ROC - NASNet Mobile com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da NASNet Mobile. Fonte: O autor (2024).

A quarta CNN a ser treinada foi a ResNet50V2. Os resultados obtidos podem ser observados na TABELA 9.

TABELA 9 – RESNET50V2

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8490	0,7350	0,7758	0,8972	0,7810	0,8076 ± 0,0578
Sensibilidade	0,8852	0,7313	0,7518	0,9660	0,8264	0,8321 ± 0,0865
Especificidade	0,7857	0,7424	0,8171	0,7463	0,6818	0,7547 ± 0,0456
Precisão	0,8780	0,8522	0,8760	0,8931	0,8500	0,8699 ± 0,0164
F-score	0,8816	0,7871	0,8092	0,9281	0,8380	0,8488 ± 0,0507
AUC	0,9082	0,8009	0,8639	0,9340	0,8920	0,8798 ± 0,0455

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 40.

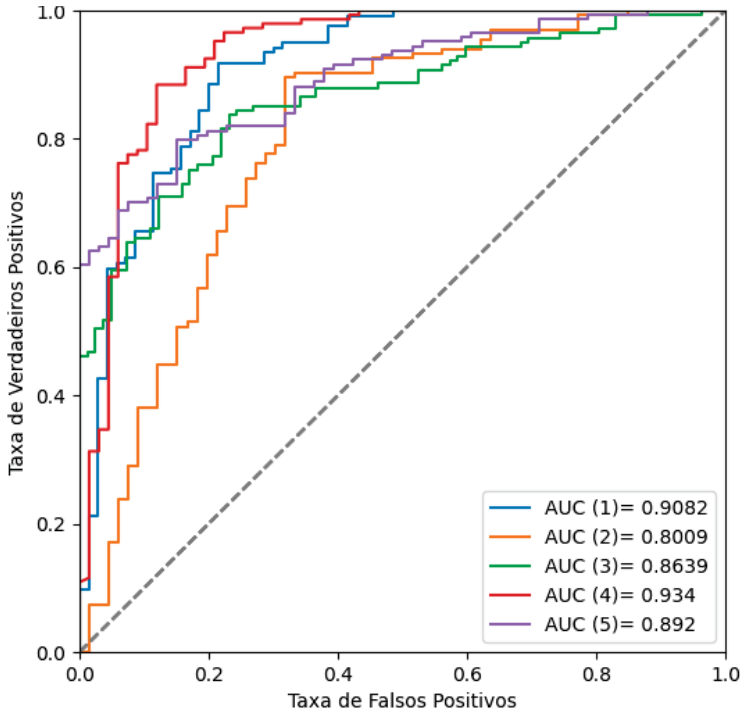


FIGURA 40 – Curvas ROC - ResNet50V2. Curvas ROC e AUC obtidas para cada treinamento da ResNet50V2. Fonte: O autor (2024).

A ResNet50V2 também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 10.

TABELA 10 – RESNET50V2 - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8177	0,6000	0,8475	0,8645	0,8667	0,7993 $\pm$ 0,1012
Sensibilidade	0,8852	0,5224	0,8865	0,9252	0,9792	0,8397 $\pm$ 0,1623
Especificidade	0,7000	0,7576	0,7805	0,7313	0,6212	0,7181 $\pm$ 0,0554
Precisão	0,8372	0,8140	0,8741	0,8831	0,8494	0,8516 $\pm$ 0,0250
F-score	0,8606	0,6364	0,8803	0,9037	0,9097	0,8381 $\pm$ 0,1024
AUC	0,8679	0,7705	0,9036	0,9249	0,9393	0,8812 $\pm$ 0,0604

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 41.

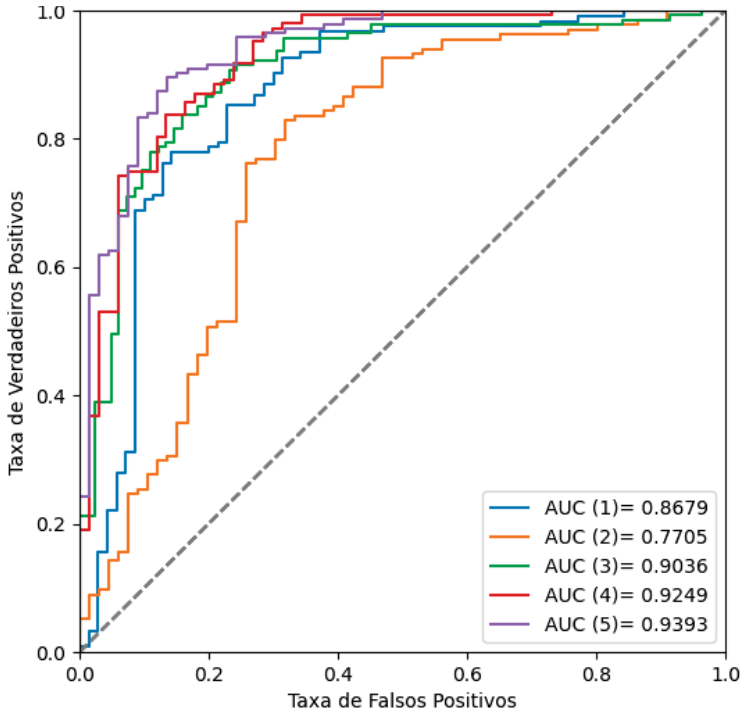


FIGURA 41 – Curvas ROC - ResNet50V2 com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da ResNet50V2. Fonte: O autor (2024).

A quinta CNN a ser treinada foi a VGG16. Os resultados obtidos podem ser observados na TABELA 11.

TABELA 11 – VGG16

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7917	0,7100	0,7399	0,8458	0,7476	0,7670 ± 0,0473
Sensibilidade	0,8197	0,7463	0,8227	0,8844	0,8819	0,8310 ± 0,0506
Especificidade	0,7429	0,6364	0,5976	0,7612	0,4545	0,6385 ± 0,1109
Precisão	0,8475	0,8065	0,7785	0,8904	0,7791	0,8204 ± 0,0431
F-score	0,8333	0,7752	0,8000	0,8874	0,8274	0,8247 ± 0,0376
AUC	0,8737	0,7529	0,7847	0,9022	0,8104	0,8248 ± 0,0554

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 42.

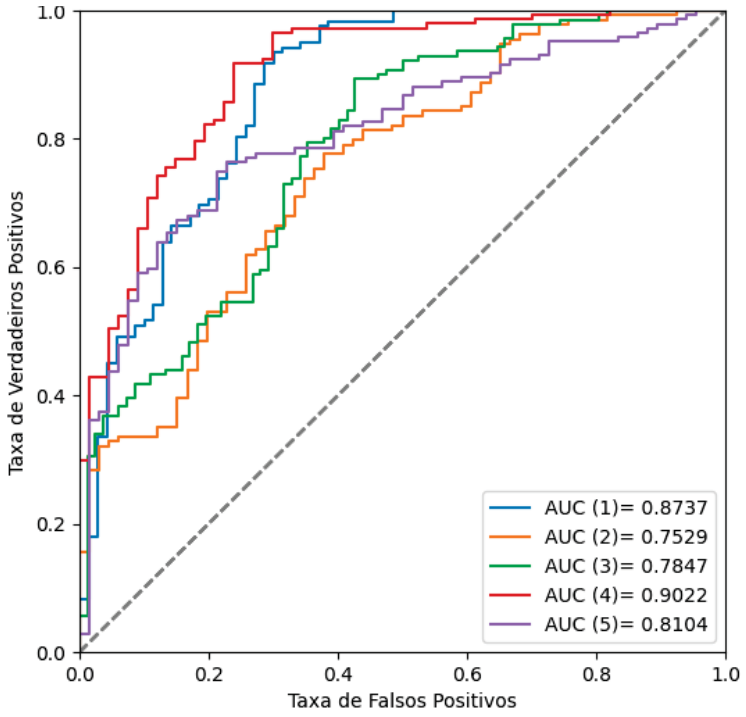


FIGURA 42 – Curvas ROC - VGG16. Curvas ROC e AUC obtidas para cada treinamento da VGG16. Fonte: O autor (2024).

A VGG16 também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 12.

TABELA 12 – VGG16 - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8281	0,7500	0,8341	0,8645	0,8143	0,8182 $\pm$ 0,0378
Sensibilidade	0,8443	0,9328	0,8298	0,9048	0,8681	0,8759 $\pm$ 0,0381
Especificidade	0,8000	0,3788	0,8415	0,7761	0,6970	0,6987 $\pm$ 0,1667
Precisão	0,8803	0,7530	0,9000	0,8986	0,8621	0,8588 $\pm$ 0,0547
F-score	0,8619	0,8333	0,8635	0,9017	0,8651	0,8651 $\pm$ 0,0217
AUC	0,8578	0,7562	0,8980	0,9092	0,8761	0,8595 $\pm$ 0,0546

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 43.

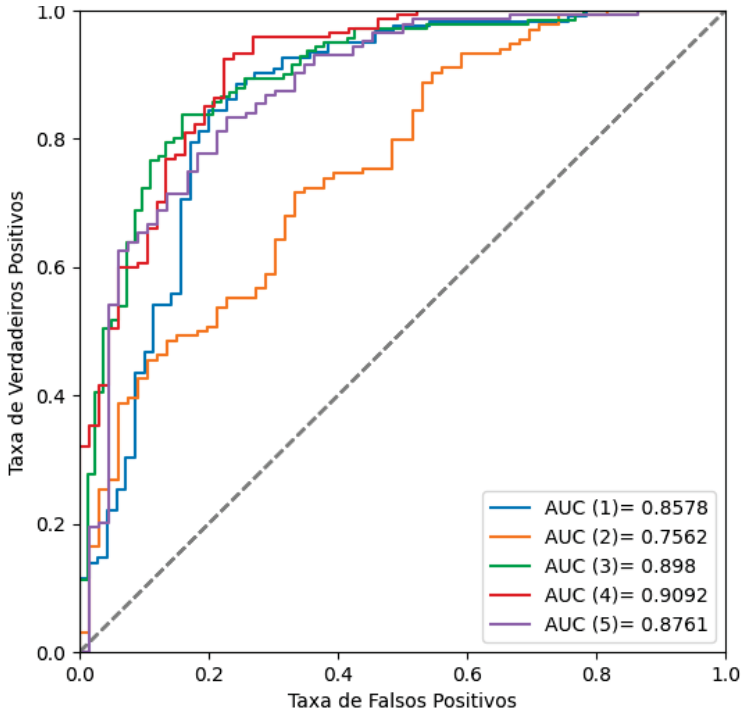


FIGURA 43 – Curvas ROC - VGG16 com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da VGG16. Fonte: O autor (2024).

A sexta CNN a ser treinada foi a VGG19. Os resultados obtidos podem ser observados na TABELA 13.

TABELA 13 – VGG19

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7865	0,7700	0,8341	0,8131	0,8143	0,8036 $\pm$ 0,0226
Sensibilidade	0,7787	0,9254	0,8794	0,9388	0,8611	0,8767 $\pm$ 0,0567
Especificidade	0,8000	0,4545	0,7561	0,5373	0,7121	0,6520 $\pm$ 0,1330
Precisão	0,8716	0,7750	0,8611	0,8166	0,8671	0,8383 $\pm$ 0,0372
F-score	0,8225	0,8435	0,8702	0,8734	0,8641	0,8548 $\pm$ 0,0192
AUC	0,8645	0,7993	0,9029	0,8641	0,8895	0,8641 $\pm$ 0,0356

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 44.

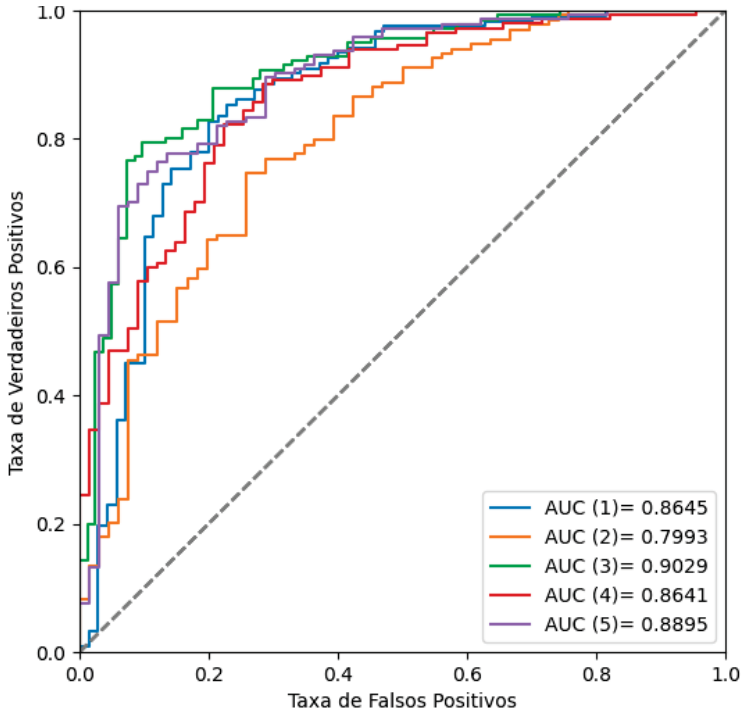


FIGURA 44 – Curvas ROC - VGG19. Curvas ROC e AUC obtidas para cada treinamento da VGG19. Fonte: O autor (2024).

A VGG19 também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 14.

TABELA 14 – VGG19 - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8490	0,7700	0,8251	0,8832	0,8667	0,8388 ± 0,0441
Sensibilidade	0,8770	0,8731	0,8723	0,9592	0,9375	0,9038 ± 0,0414
Especificidade	0,8000	0,5606	0,7439	0,7164	0,7121	0,7066 ± 0,0888
Precisão	0,8843	0,8014	0,8542	0,8813	0,8766	0,8595 ± 0,0346
F-score	0,8807	0,8357	0,8632	0,9186	0,9060	0,8808 ± 0,0332
AUC	0,8991	0,7924	0,9069	0,9335	0,9154	0,8894 ± 0,0557

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 45.

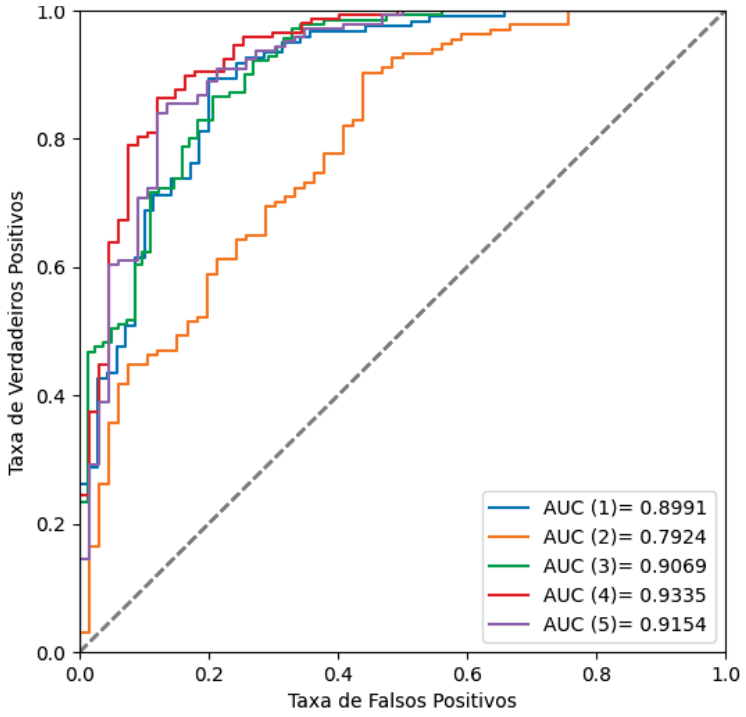


FIGURA 45 – Curvas ROC - VGG19 com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da VGG19. Fonte: O autor (2024).

A sétima CNN a ser treinada foi a Xception. Os resultados obtidos podem ser observados na TABELA 15.

TABELA 15 – XCEPTION

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7813	0,7350	0,7713	0,8785	0,8524	0,8037 ± 0,0534
Sensibilidade	0,7705	0,7761	0,7163	0,8980	0,8472	0,8016 ± 0,0637
Especificidade	0,8000	0,6515	0,8659	0,8358	0,8636	0,8034 ± 0,0796
Precisão	0,8704	0,8189	0,9018	0,9231	0,9313	0,8891 ± 0,0409
F-score	0,8174	0,7969	0,7984	0,9103	0,8873	0,8421 ± 0,0474
AUC	0,8785	0,7602	0,8591	0,9102	0,9274	0,8671 ± 0,0585

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 46.

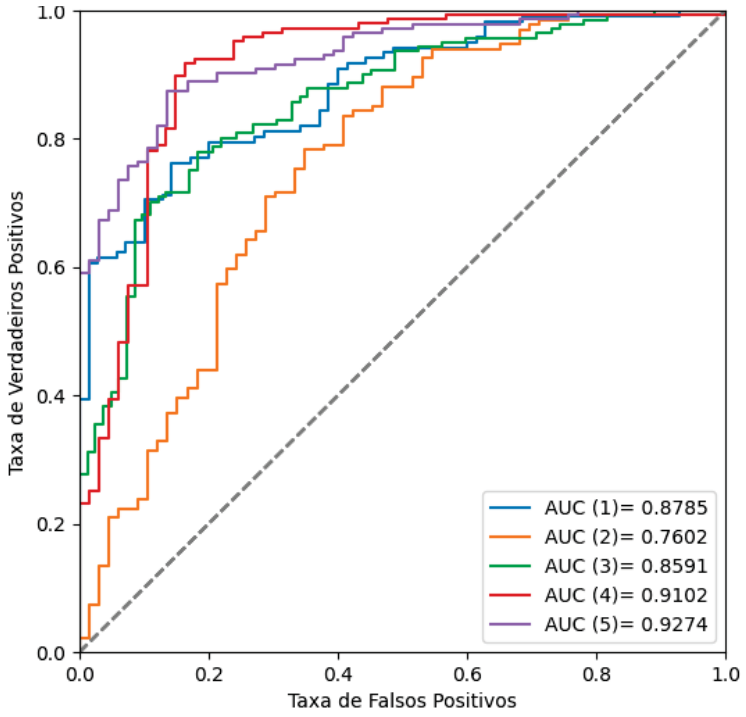


FIGURA 46 – Curvas ROC - Xception. Curvas ROC e AUC obtidas para cada treinamento da Xception. Fonte: O autor (2024).

A Xception também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 16.

TABELA 16 – XCEPTION - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7969	0,7150	0,7623	0,8738	0,8333	0,7963 ± 0,0550
Sensibilidade	0,8852	0,7313	0,7021	0,8912	0,8819	0,8184 ± 0,0835
Especificidade	0,6429	0,6818	0,8659	0,8358	0,7273	0,7507 ± 0,0865
Precisão	0,8120	0,8235	0,9000	0,9225	0,8759	0,8668 ± 0,0428
F-score	0,8471	0,7747	0,7888	0,9066	0,8789	0,8392 ± 0,0507
AUC	0,8794	0,7992	0,8483	0,9133	0,9096	0,8700 ± 0,0425

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 47.

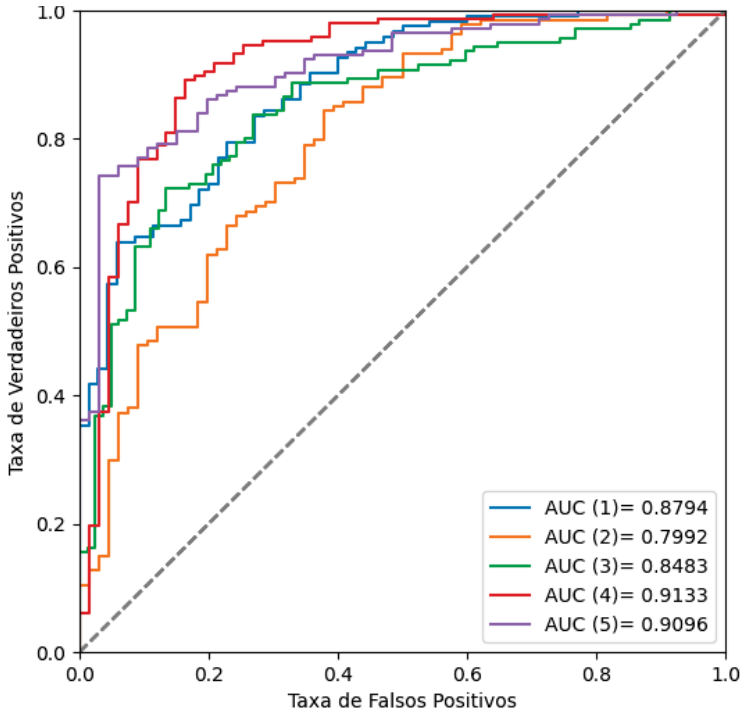


FIGURA 47 – Curvas ROC - Xception com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento da Xception. Fonte: O autor (2024).

Entre os modelos de CNNs, o qual atingiu a melhor acurácia foi a VGG19 com aumento de dados, com **0,8388** ± 0,0441. Na sequência, em segundo e terceiro lugares, ficaram as redes NASNet Mobile e VGG16, ambas com aumento de dados, com 0,8186 ± 0,027 e 0,8182 ± 0,0423, respectivamente. Conforme pode ser observado na TABELA 2, existe um desbalanceamento entre as classes, sendo os casos de paciente ”normais” mais frequentes no conjunto de dados do que os pacientes que apresentam aumento do átrio esquerdo. Sendo assim, a métrica da acurácia deve ser avaliada cuidadosamente, uma vez que pode oferecer uma visão distorcida da assertividade das redes.

Para a métrica de sensibilidade, a CNN que atingiu o melhor resultado – mostrando uma maior capacidade de detectar verdadeiros positivos – foi a VGG19 com aumento de dados, com **0,9038** $\pm$ 0,0414. Na sequência, em segundo e terceiro lugares, ficaram as redes NASNet Mobile com aumento de dados e VGG19, com 0,8976 $\pm$ 0,0519 e 0,8767 $\pm$ 0,0634, respectivamente.

Para a métrica de especificidade, a CNN que atingiu o melhor resultado – mostrando uma maior capacidade de detectar casos verdadeiros negativos – foi a InceptionV3, com **0,8478** $\pm$ 0,1065. Na sequência, em segundo e terceiro lugares, ficaram as redes InceptionV3 com aumento de dados e Xception, com 0,8176 $\pm$ 0,1087 e 0,8034 $\pm$ 0,0890, respectivamente.

Para a métrica de precisão, a CNN que atingiu o melhor resultado – mostrando uma maior capacidade de acertar os casos previstos por ela como positivos – foi a InceptionV3 com aumento de dados, com **0,9051** $\pm$ 0,0408. Na sequência, em segundo e terceiro lugares, ficaram as redes InceptionV3 e Xception, com 0,9028 $\pm$ 0,0606 e 0,8891 $\pm$ 0,0458, respectivamente.

Para a métrica de F-score, a CNN que atingiu o melhor resultado foi a VGG19 com aumento de dados, com **0,8808** $\pm$ 0,0332. Na sequência, em segundo e terceiro lugares, ficaram as redes NASNet Mobile e VGG16, ambas com aumento de dados, com 0,8673 $\pm$ 0,0198 e 0,8651 $\pm$ 0,0243, respectivamente. Como existe um desbalanceamento de classes, esta métrica oferece um panorama melhor para entender a assertividade das redes do que a acurácia. Além disso, por ser composta pela precisão e sensibilidade, ela é diretamente impactada pelo equilíbrio entre estas duas, fornecendo um parâmetro de como os modelos são capazes de detectar verdadeiros positivos ao passo que estejam reduzindo a possibilidade de resultados falsos positivos.

Para a métrica de AUC, a CNN que atingiu o melhor resultado – ou seja, tem uma capacidade maior de distinguir as duas classes do problema – foi a InceptionV3 com aumento de dados, com **0,8976** $\pm$ 0,0383. Na sequência, em segundo e terceiro lugares, ficaram as redes VGG19 com aumento de dados e DenseNet-121, com 0,8894 $\pm$ 0,0557 e 0,8877 $\pm$ 0,0610, respectivamente.

Estes resultados corroboram com a revisão da literatura (Capítulo 3) na qual destacou-se que as CNNs são uma estratégia viável para realizar a classificação de doenças em radiografias de animais. Além disso, na maioria dos casos, a técnica de aumento de dados foi capaz de promover uma melhoria geral nas métricas, mostrando que esses modelos se beneficiaram de uma maior diversidade de dados.

5.2 MODELOS BASEADOS EM VITS

Para estabelecer um comparativo com as CNNs, foram treinados dois modelos que implementam ViTs. O primeiro diz respeito a um ViT em sua forma original, enquanto o segundo adiciona a ele as técnicas de Shifted Patch Tokenization (SPT) e Locality Self-Attention (LSA). Neste segundo caso, o modelo será denotado pelas siglas ”SL” no início do nome. Ambos os modelos foram treinados com e sem o auxílio da técnica de aumento de dados.

A primeira rede treinada aplica a estrutura original do ViT. Os resultados obtidos podem ser observados na TABELA 17.

TABELA 17 – VIT

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7292	0,6650	0,7220	0,7757	0,7762	0,7336 ± 0,0411
Sensibilidade	0,8525	0,8433	0,9574	0,8163	0,9444	0,8828 ± 0,0570
Especificidade	0,5143	0,3030	0,3171	0,6866	0,4091	0,4460 ± 0,1421
Precisão	0,7536	0,7107	0,7068	0,8511	0,7771	0,7599 ± 0,0527
F-score	0,8000	0,7713	0,8133	0,8333	0,8527	0,8141 ± 0,0279
AUC	0,7369	0,6427	0,7090	0,8558	0,8642	0,7617 ± 0,0859

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 48.

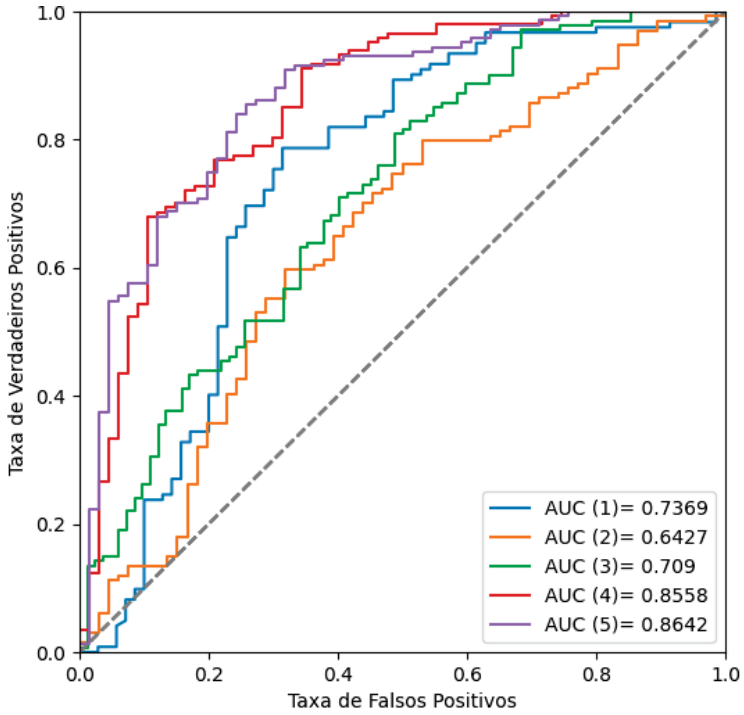


FIGURA 48 – Curvas ROC - ViT. Curvas ROC e AUC obtidas para cada treinamento do ViT. Fonte: O autor (2024).

O ViT também foi utilizada aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 18.

TABELA 18 – VIT - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,6719	0,6800	0,6413	0,6869	0,7476	0,6855 $\pm$ 0,0347
Sensibilidade	0,9590	0,9925	0,9433	1,0000	0,9931	0,9776 $\pm$ 0,0223
Especificidade	0,1714	0,0455	0,1220	0,0000	0,2121	0,1102 $\pm$ 0,0782
Precisão	0,6686	0,6786	0,6488	0,6869	0,7333	0,6832 $\pm$ 0,0281
F-score	0,7879	0,8061	0,7688	0,8144	0,8437	0,8042 $\pm$ 0,0252
AUC	0,7363	0,5979	0,7256	0,8455	0,8702	0,7551 $\pm$ 0,0973

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 49.

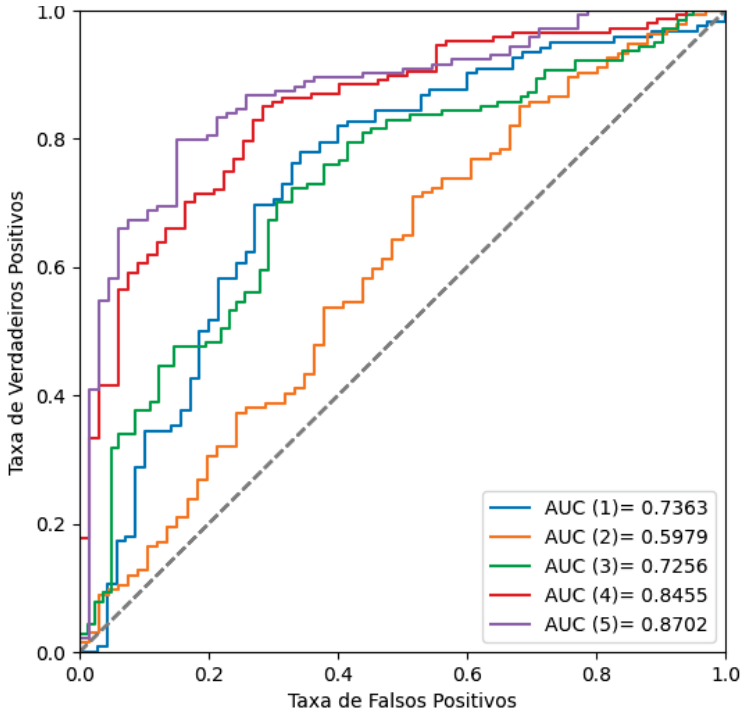


FIGURA 49 – Curvas ROC - ViT com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento do ViT. Fonte: O autor (2024).

A segunda rede treinada aplica as técnicas de Shifted Patch Tokenization (SPT) e Locality Self-Attention (LSA) à estrutura original do ViT. Os resultados obtidos podem ser observados na TABELA 19.

TABELA 19 – SL-VIT

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,7292	0,6750	0,6771	0,7664	0,7381	0,7171 $\pm$ 0,0399
Sensibilidade	0,8852	0,8507	0,7943	0,9864	0,7708	0,8575 $\pm$ 0,0850
Especificidade	0,4571	0,3182	0,4756	0,2836	0,6667	0,4402 $\pm$ 0,1519
Precisão	0,7397	0,7170	0,7226	0,7513	0,8346	0,7530 $\pm$ 0,0476
F-score	0,8060	0,7782	0,7568	0,8529	0,8014	0,7991 $\pm$ 0,0360
AUC	0,7463	0,6882	0,6958	0,8563	0,8413	0,7656 $\pm$ 0,0794

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 50.

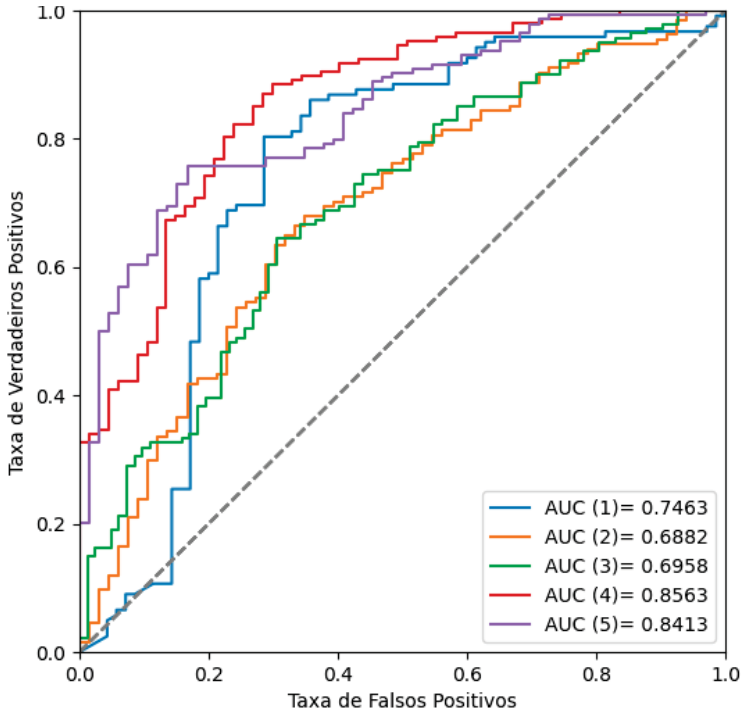


FIGURA 50 – Curvas ROC - SL-ViT. Curvas ROC e AUC obtidas para cada treinamento do SL-ViT. Fonte: O autor (2024).

O ViT com SL também foi utilizado aplicando-se a técnica de aumento de dados. Os resultados obtidos podem ser observados na TABELA 20.

TABELA 20 – SL-VIT - AUMENTO DE DADOS

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,6354	0,6650	0,6413	0,7664	0,7000	0,6816 ± 0,0538
Sensibilidade	1,0000	0,8507	1,0000	0,8435	1,0000	0,9389 ± 0,0838
Especificidade	0,0000	0,2879	0,0244	0,5970	0,0455	0,1909 ± 0,2548
Precisão	0,6354	0,7081	0,6380	0,8212	0,6957	0,6997 ± 0,0755
F-score	0,7771	0,7729	0,7790	0,8322	0,8205	0,7963 ± 0,0278
AUC	0,7664	0,6067	0,6919	0,8081	0,8069	0,7360 ± 0,0863

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos *folds*, as quais são apresentadas na FIGURA 51.

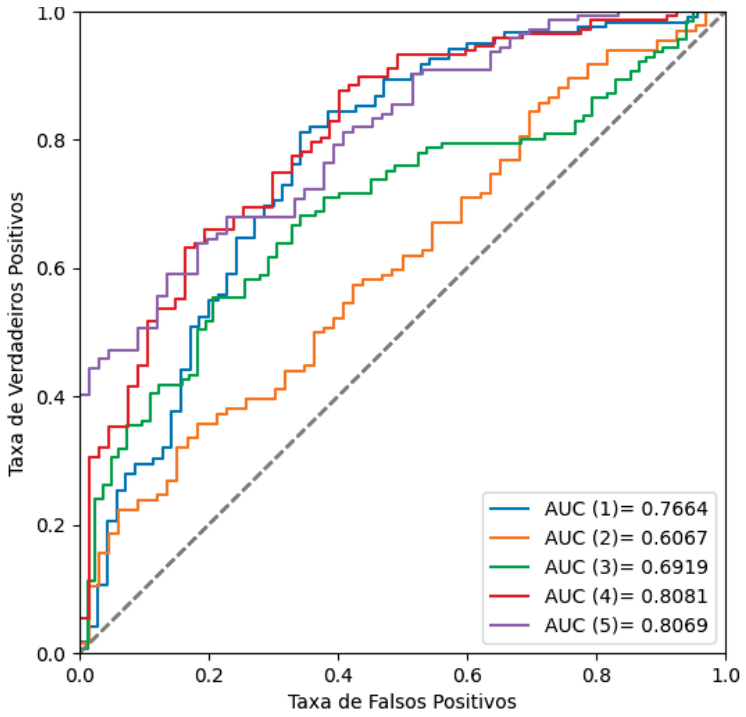


FIGURA 51 – Curvas ROC - SL-ViT com aumento de dados. Curvas ROC e AUC obtidas para cada treinamento do SL-ViT. Fonte: O autor (2024).

O ViT com melhor acurácia foi a arquitetura original, sem aumento de dados e sem SL, com **0,7336** ± 0,0460 – um resultado aquém dos obtidos pelas CNNs.

O ViT com melhor desempenho de sensibilidade foi a arquitetura original com aumento de dados e sem SL, com **0,9776** ± 0,0249 – neste caso superando às CNNs. Desta forma, é possivel dizer que o ViT demonstrou uma capacidade maior do que as CNNs de detectar verdadeiros positivos. Contudo, ao observar a especificidade baixa desta mesma rede (0,4460 ± 0.1589), nota-se que esta tem pouca capacidade de detectar os verdadeiros negativos, ou seja, o ViT tende a pressupor que todos os casos são positivos, gerando assim um elevado número de falsos positivos.

Ainda sobre especificidade, ViT com melhor desempenho nesta métrica foi a arquitetura original e sem SL, com **0,4460** $\pm$ 0.1589 – conforme mencionado anteriormente, um resultado significativamente inferior ao das CNNs. Isto é, as CNNs se mostraram superiores para detectar casos verdadeiros negativos, gerando quantidades menores de falsos positivos.

O ViT com a arquitetura original e sem SL também resultou na melhor precisão, com **0,7599** $\pm$ 0,0589 – também um resultado aquém dos obtidos pelas CNNs. Isto é, as CNNs se mostraram superiores para detectar casos verdadeiros positivos, ao mesmo tempo gerando uma menor taxa de falsos positivos. Ou seja, as CNNs não tenderam a apenas pressupor que todos os casos são positivos, como aconteceu com os ViTs.

A arquitetura original do ViT e sem SL também resultou no melhor F-score, com **0,8141** $\pm$ 0,0312 – um resultado inferior aos obtidos pelas CNNs.

Por fim, o ViT com melhor desempenho de AUC foi a arquitetura com SL e sem aumento de dados, com **0,7656** $\pm$ 0,0794 – um resultado inferior aos obtidos pelas CNNs. Isto implica que as CNNs mostraram uma capacidade superior à dos VITs de distinguir as duas classes do problema.

Estes resultados corroboram com o que é frequentemente visto na literatura: o ViTs alcançam desempenho notável em grandes conjuntos de dados, mas tendem a ter um desempenho inferior ao das CNNs quando treinados do zero em conjuntos de dados menores (AKKAYA et al., 2024).

Além disso, as técnicas de *Shifted Patch Tokenization* e *Locality Self-Attention Mechanism* não foram capazes de promover melhorias para o cenário de conjunto de dados reduzidos deste trabalho. Dado que as CNNs se concentram principalmente na extração de características locais, o que pode limitar sua capacidade de compreender o contexto global e impactar sua performance em tarefas como a compreensão de cenas ou geração de legendas para imagens (MAURÍCIO; DOMINGUES; BERNARDINO, 2023), neste caso em que a doença se manifesta em uma região específica da anatomia do animal esta característica pode ter sido vantajosa, sendo um fator que contribuiu para os resultados superiores obtidos pelas CNNs em relação aos ViTs.

5.3 ENSEMBLES

Com os resultados obtidos na seção 5.2 e seção 5.1, foram gerados três ensembles. Como exemplificado na FIGURA 52, os melhores modelos treinados, incluindo suas camadas de Global Average Pooling 2D, Dense + reLu e Dense + sigmoid, foram colocados em paralelo, com uma camada de Average ao final para realizar a média dos valores de saída dos sigmoids. Desta forma, a informação de uma imagem de entrada passa igualmente por todas as redes ao mesmo tempo até ser combinada na última camada.

A primeira combinação de modelos, **Ensemble-A**, foi construída utilizando as duas CNNs que apresentaram os melhores resultados de **AUC**, sendo elas InceptionV3 e VGG19, ambas com aumento de dados.

A segunda combinação de modelos, **Ensemble-B**, foi construída utilizando as três CNNs que apresentaram os melhores resultados de **F-score**, sendo elas VGG16, VGG19, e NASNet Mobile, todas com aumento de dados.

A terceira combinação de modelos, **Ensemble-C**, foi construída utilizando as três CNNs que apresentaram os melhores resultados de **AUC**, sendo elas InceptionV3, VGG19 e DenseNet-121, todas com aumento de dados. No caso desta última, o desempenho sem e com aumento de dados foi praticamente o mesmo (0,8877 $\pm$ 0,0610 vs 0,8876 $\pm$ 0,0580). Sendo assim, foi escolhida a versão com aumento de dados, que viu uma variabilidade maior de dados ao longo de seu treinamento.

Todos os ensembles foram aplicados nos 5 *folds*, gerando as métricas de acurácia, sensibilidade, especificidade, precisão, F-score e AUC.

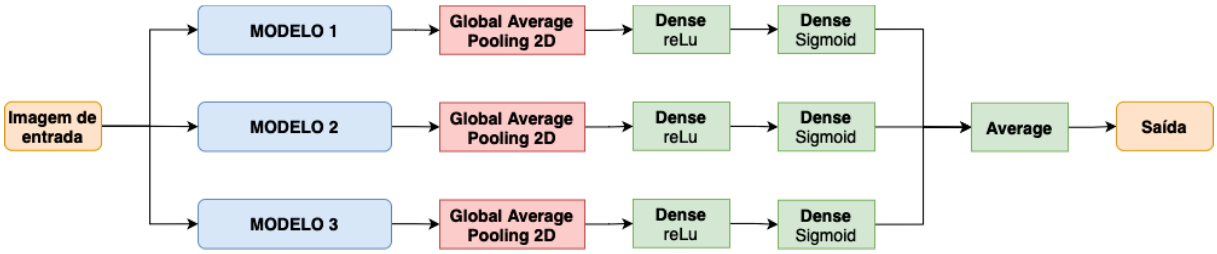


FIGURA 52 – Configuração dos ensembles. Os modelos treinados processam a mesma imagem de entrada de maneira paralela, adicionando uma camada para realizar a média entre as saídas dos sigmoides. Fonte: O autor (2024).

Os resultados obtidos para o Ensemble-A podem ser visualizados na TABELA 21.

TABELA 21 – ENSEMBLE-A

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8177	0,8000	0,8251	0,8879	0,8619	0,8385 $\pm$ 0,0356
Sensibilidade	0,8197	0,8955	0,8369	0,9524	0,8958	0,8801 $\pm$ 0,0530
Especificidade	0,8143	0,6061	0,8049	0,7463	0,7879	0,7519 $\pm$ 0,0856
Precisão	0,8850	0,8219	0,8806	0,8917	0,9021	0,8763 $\pm$ 0,0314
F-score	0,8511	0,8571	0,8582	0,9211	0,8990	0,8773 $\pm$ 0,0310
AUC	0,9107	0,8362	0,9172	0,9467	0,9303	0,9082 $\pm$ 0,0426

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos folds, as quais são apresentadas na FIGURA 53.

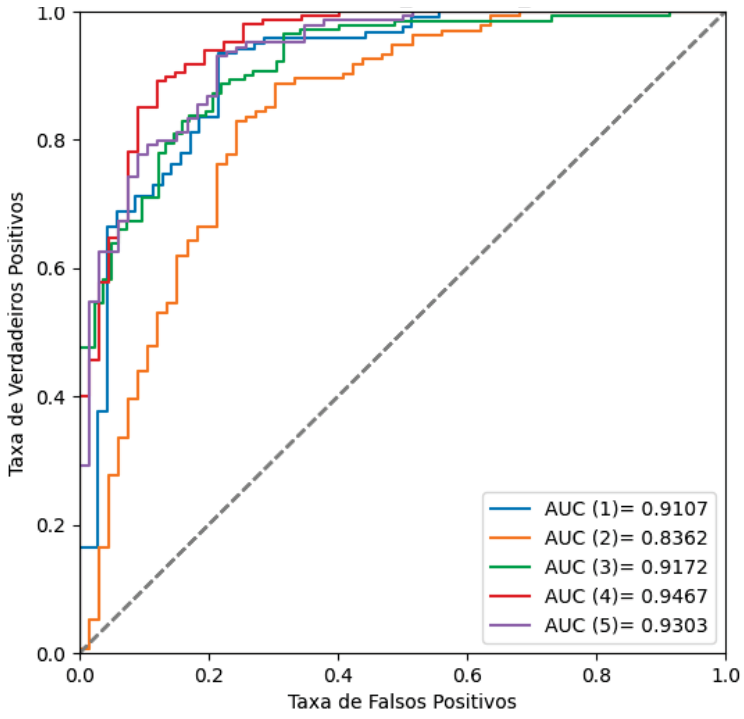


FIGURA 53 – Curvas ROC - Ensemble-A (InceptionV3 + VGG19 (DA)). Curvas ROC e AUC obtidas para cada treinamento do ensemble. Fonte: O autor (2024).

Os resultados obtidos para o Ensemble-B podem ser visualizados na TABELA 22.

TABELA 22 – ENSEMBLE-B

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8438	0,7800	0,8386	0,8879	0,8571	0,8415 ± 0,0393
Sensibilidade	0,8934	0,9030	0,8511	0,9456	0,9583	0,9103 ± 0,0430
Especificidade	0,7571	0,5303	0,8171	0,7612	0,6364	0,7004 ± 0,1157
Precisão	0,8651	0,7961	0,8889	0,8968	0,8519	0,8597 ± 0,0399
F-score	0,8790	0,8462	0,8696	0,9205	0,9020	0,8834 ± 0,0288
AUC	0,8902	0,7880	0,9182	0,9299	0,9145	0,8881 ± 0,0578

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos folds, as quais são apresentadas na FIGURA 54.

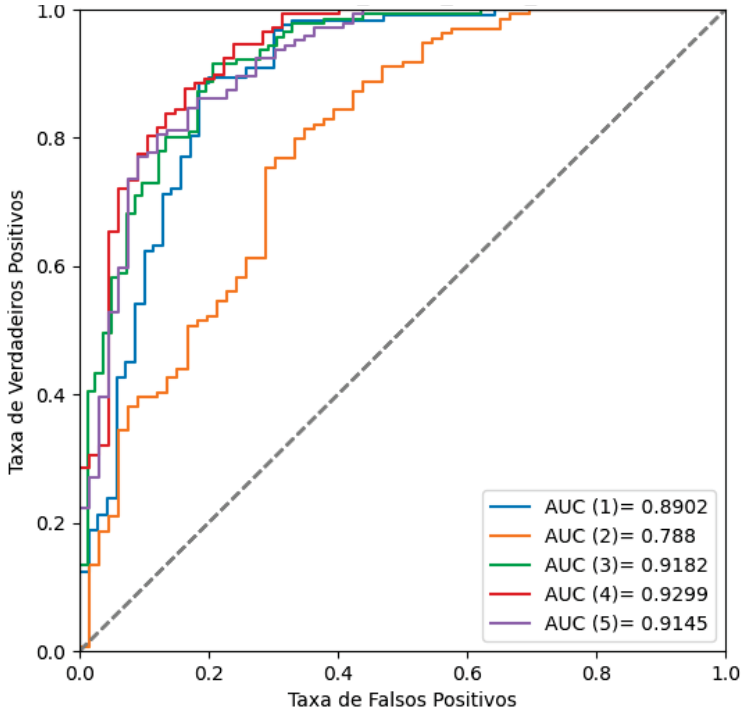


FIGURA 54 – Curvas ROC - Ensemble-B (VGG16 (DA) + VGG19 (DA) + NASNet (DA)). Curvas ROC e AUC obtidas para cada treinamento do ensemble. Fonte: O autor (2024).

Os resultados obtidos para o Ensemble-C podem ser visualizados na TABELA 23.

TABELA 23 – ENSEMBLE-C

	Fold 01	Fold 02	Fold 03	Fold 04	Fold 05	Média
Acurácia	0,8333	0,7950	0,8520	0,9206	0,8714	0,8545 $\pm$ 0,0465
Sensibilidade	0,8443	0,8582	0,8723	0,9660	0,9097	0,8901 $\pm$ 0,0489
Especificidade	0,8143	0,6667	0,8171	0,8209	0,7879	0,7814 $\pm$ 0,0654
Precisão	0,8879	0,8394	0,8913	0,9221	0,9034	0,8888 $\pm$ 0,0307
F-score	0,8655	0,8487	0,8817	0,9435	0,9066	0,8892 $\pm$ 0,0371
AUC	0,9056	0,8245	0,9290	0,9537	0,9366	0,9099 $\pm$ 0,0508

FONTE: O autor (2024).

Na sequência, foram geradas as curvas ROC para cada um dos folds, as quais são apresentadas na FIGURA 55.

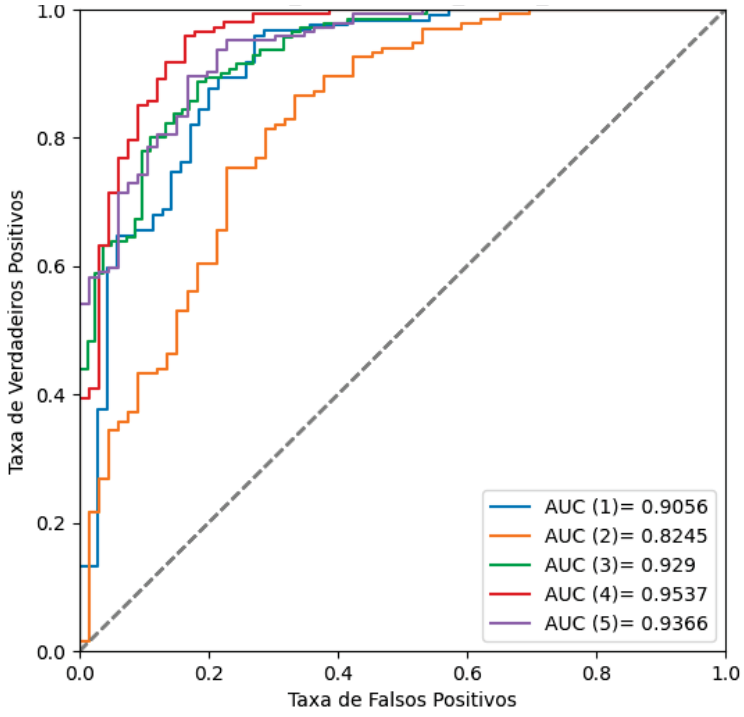


FIGURA 55 – Curvas ROC - Ensemble-C (InceptionV3 (DA) + VGG19 (DA) + DenseNet-121 (DA)). Curvas ROC e AUC obtidas para cada treinamento do ensemble. Fonte: O autor (2024).

Os ensembles perderam nas métricas de sensibilidade, na qual o ViT com aumento de dados obteve 0,9776 $\pm$ 0,0249 vs 0,9103 $\pm$ 0,0430 do Ensemble-B – embora já tenha sido discutido na seção anterior o problema com este modelo; na especificidade, na qual a InceptionV3 obteve **0,8478** $\pm$ 0,1065 vs 0,7814 $\pm$ 0,0654 do Ensemble-C (melhor resultado do grupo); e na precisão, na qual InceptionV3 com aumento de dados obteve **0,9051** $\pm$ 0,0408 vs 0.8888 $\pm$ 0.0371 do Ensemble-C (melhor resultado do grupo). O Ensemble-C obteve o melhor resultado geral nas métricas de acurácia (**0,8545** $\pm$ 0,0465), F-score (**0,8892** $\pm$ 0,0371) e AUC (**0,9099** $\pm$ 0,050), perdendo para o Ensemble-B na sensibilidade (0,8901 $\pm$ 0,048 vs **0,9103** $\pm$ 0,0430, respectivamente). Isto é, o Ensemble-C resultou em um modelo mais bem equilibrado na detecção de verdadeiros positivos, minimizando os resultados falsos positivos. Além disso, também resultou em um classificador binário que superou todos os modelos treinados.

5.4 COMPILADO DOS RESULTADOS

Com o objetivo de facilitar a visualização dos resultados, as médias das métricas obtidas na seção anterior foram compiladas em duas tabelas. A primeira, TABELA 24 (Comparativo A), traz as médias e desvios-padrão para a acurácia, sensibilidade e especificidade. A segunda, TABELA 25 (Comparativo B), traz as médias e desvios-padrão para a precisão, F-score e AUC.

Em ambas as tabelas os modelos estão elecandos em três grupos, sendo eles os modelos baseados em CNNs, os modelos baseados em ViTs, e os ensembles. Quando o nome

do modelo é seguido pela sigla ”DA” significa que este foi treinado usando aumento de dados. O melhor resultado obtido para cada coluna é apresentado em negrito.

TABELA 24 – COMPARATIVO A

Modelo	Acurácia	Sensibibilidade	Especificidade
DenseNet-121	0,8142 ± 0,0267	0,8519 ± 0,0869	0,7337 ± 0,1749
DenseNet-121 (DA)	0,8134 ± 0,0473	0,8385 ± 0,0747	0,7712 ± 0,0920
InceptionV3	0,7510 ± 0,0528	0,7033 ± 0,0816	**0,8478** ± 0,1065
InceptionV3 (DA)	0,8118 ± 0,0451	0,8025 ± 0,1137	0,8176 ± 0,1087
NASNet Mobile	0,7664 ± 0,0845	0,7447 ± 0,1480	0,7962 ± 0,1233
NASNet Mobile (DA)	0,8186 ± 0,0270	0,8976 ± 0,0519	0,6593 ± 0,1188
ResNet50V2	0,8076 ± 0,0647	0,8321 ± 0,0967	0,7547 ± 0,0510
ResNet50V2 (DA)	0,7993 ± 0,1131	0,8397 ± 0,1815	0,7181 ± 0,0619
VGG16	0,7670 ± 0,0529	0,8310 ± 0,0566	0,6385 ± 0,1240
VGG16 (DA)	0,8182 ± 0,0423	0,8759 ± 0,0426	0,6987 ± 0,1864
VGG19	0,8036 ± 0,0253	0,8767 ± 0,0634	0,6520 ± 0,1487
VGG19 (DA)	0,8388 ± 0,0441	0,9038 ± 0,0414	0,7066 ± 0,0888
Xception	0,8037 ± 0,0597	0,8016 ± 0,0712	0,8034 ± 0,0890
Xception (DA)	0,7963 ± 0,0615	0,8184 ± 0,0934	0,7507 ± 0,0967
ViT	0,7336 ± 0,0460	0,8828 ± 0,0638	0,4460 ± 0.1589
ViT (DA)	0,6855 ± 0,0388	**0,9776** ± 0,0249	0,1102 ± 0,0875
SL-ViT	0,7171 ± 0,0399	0,8575 ± 0,0850	0,4402 ± 0,1519
SL-ViT (DA)	0,6816 ± 0,0538	0,9389 ± 0,0838	0,1909 ± 0,2548
Ensemble-A	0,8385 ± 0,0356	0,8801 ± 0,0530	0,7519 ± 0,0856
Ensemble-B	0,8415 ± 0,0393	0,9103 ± 0,0430	0,7004 ± 0,1157
Ensemble-C	**0,8545** ± 0,0465	0,8901 ± 0,0489	0,7814 ± 0,0654

FONTE: O autor (2024).

TABELA 25 – COMPARATIVO B

Modelo	Precisão	F-score	AUC
DenseNet-121	0,8712 ± 0,0585	0,8571 ± 0,0298	0,8877 ± 0,0610
DenseNet-121 (DA)	0,8774 ± 0,0497	0,8552 ± 0,0400	0,8876 ± 0,0580
InceptionV3	0,9028 ± 0,0606	0,7871 ± 0,0521	0,8748 ± 0,0602
InceptionV3 (DA)	**0,9051** ± 0,0408	0,8452 ± 0,0530	0,8976 ± 0,0383
NASNet Mobile	0,8864 ± 0,0593	0,8010 ± 0,0904	0,8363 ± 0,0913
NASNet Mobile (DA)	0,8420 ± 0,0396	0,8673 ± 0,0198	0,8585 ± 0,0512
ResNet50V2	0,8699 ± 0,0184	0,8488 ± 0,0567	0,8798 ± 0,0509
ResNet50V2 (DA)	0,8516 ± 0,0280	0,8381 ± 0,1145	0,8812 ± 0,0675
VGG16	0,8204 ± 0,0482	0,8247 ± 0,0420	0,8248 ± 0,0620
VGG16 (DA)	0,8588 ± 0,0611	0,8651 ± 0,0243	0,8595 ± 0,0610
VGG19	0,8383 ± 0,0416	0,8548 ± 0,0214	0,8641 ± 0,0398
VGG19 (DA)	0,8595 ± 0,0346	0,8808 ± 0,0332	0,8894 ± 0,0557
Xception	0,8891 ± 0,0458	0,8421 ± 0,0530	0,8671 ± 0,0654
Xception (DA)	0,8668 ± 0,0479	0,8392 ± 0,0567	0,8700 ± 0,0475
ViT	0,7599 ± 0,0589	0,8141 ± 0,0312	0,7617 ± 0,0961
ViT (DA)	0,6832 ± 0,0314	0,8042 ± 0,0282	0,7551 ± 0,1088
SL-ViT	0,7530 ± 0,0476	0,7991 ± 0,0360	0,7656 ± 0,0794
SL-ViT (DA)	0,6997 ± 0,0755	0,7963 ± 0,0278	0,7360 ± 0,0863
Ensemble-A	0,8763 ± 0,0314	0,8773 ± 0,0310	0,9082 ± 0,0426
Ensemble-B	0,8597 ± 0,0399	0,8834 ± 0,0288	0,8881 ± 0,0578
Ensemble-C	0,8888 ± 0,0371	**0.,8892** ± 0,0371	**0,9099** ± 0,0508

FONTE: O autor (2024).

5.5 VISUALIZAÇÃO DOS RESULTADOS COM GRAD-CAM

Para permitir uma compreensão visual dos modelos que compõem o ensemble que atingiu o melhor resultado (Ensemble-C), foi aplicado o algoritmo de Mapeamento de Ativação de Classe Ponderado por Gradiente (do inglês, Gradient-weighted Class Activation Mapping - Grad-CAM) (SELVARAJU et al., 2020). Esta técnica permite produzir ”explicações visuais” dos modelos de CNNs, tornando-os mais transparentes. O Grad-CAM utiliza os gradientes calculados durante a etapa de retropropagação para gerar mapas de calor sobre a imagem de entrada, nos quais destacam-se as regiões da última camada convolucional que mais contribuíram para a tomada de decisão ao realizar a classificação.

As imagens a seguir, da FIGURA 56 à FIGURA 59, apresentam os mapas de calor resultantes do Grad-CAM, os quais foram sobrepostos às imagens de entrada para destacar visualmente as regiões mais importantes para a tomada de decisão dos modelos que compõem o Ensemble-C (DenseNet-121, InceptionV3 e VGG19). Na escala de cor utilizada, os tons mais avermelhados são as regiões nas quais os modelos mais focaram, enquanto os tons mais azulados são as regiões que tiveram menos peso na tomada de decisão.

A FIGURA 56, a seguir, exemplifica um caso ”normal” corretamente classificado pelo Ensemble-C.

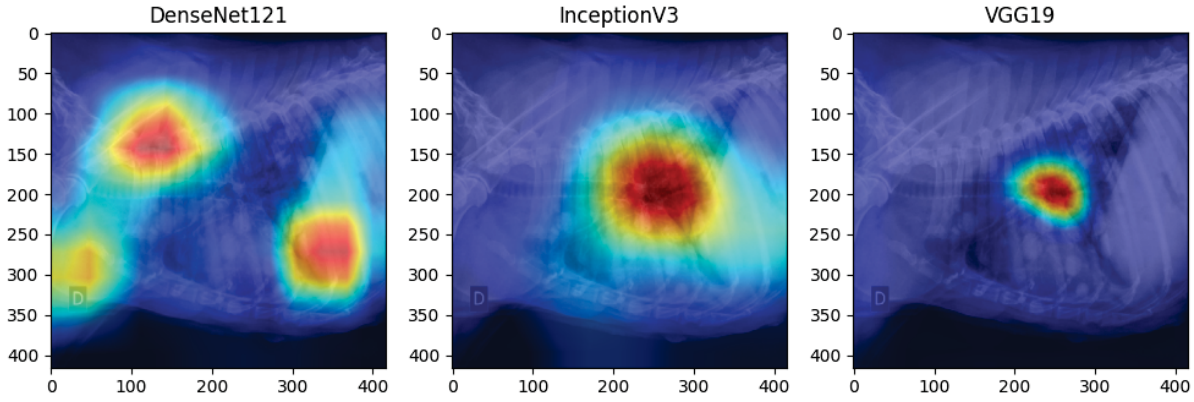


FIGURA 56 – Visualização de inferência com Grad-CAM. Classe ”normal” corretamente identificada. Fonte: O autor (2024).

A FIGURA 57, a seguir, exemplifica um caso ”normal” erroneamente classificado pelo Ensemble-C como ”aumento do átrio esquerdo”.

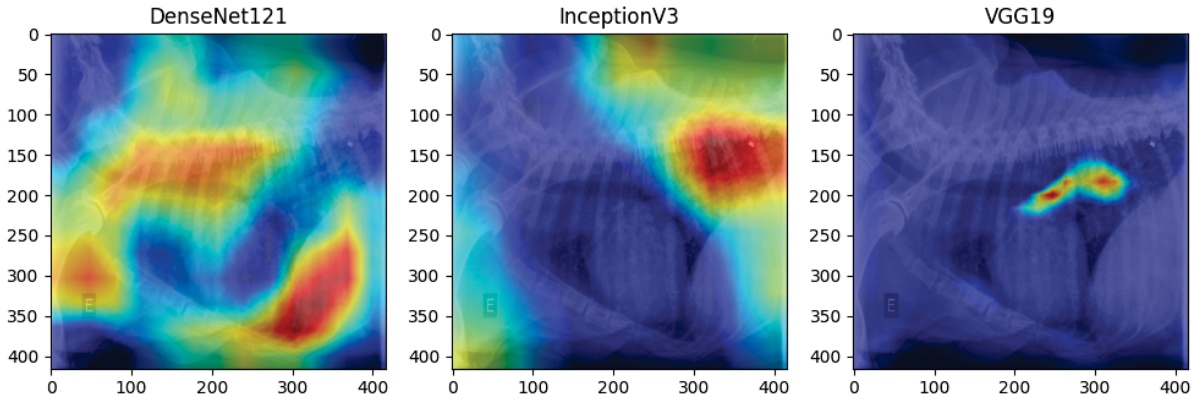


FIGURA 57 – Visualização de inferência com Grad-CAM. Classe ”normal” incorretamente identificada como ”aumento do átrio esquerdo”. Fonte: O autor (2024).

A FIGURA 58, a seguir, exemplifica um caso de ”aumento do átrio esquerdo” corretamente classificado pelo Ensemble-C.

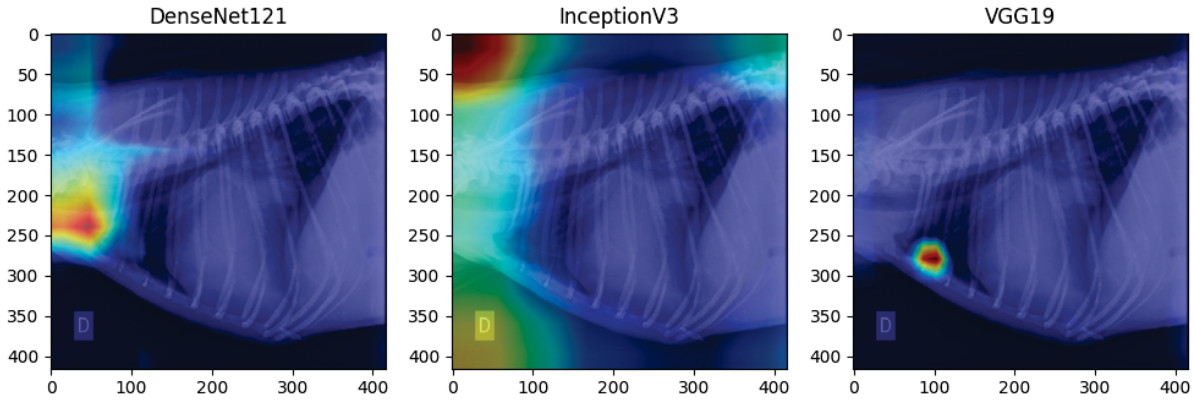


FIGURA 58 – Visualização de inferência com Grad-CAM. Classe ”aumento do átrio esquerdo” corretamente identificada. Fonte: O autor (2024).

A FIGURA 59, a seguir, exemplifica um caso de ”aumento do átrio esquerdo” erroneamente classificado pelo Ensemble-C como ”normal”.

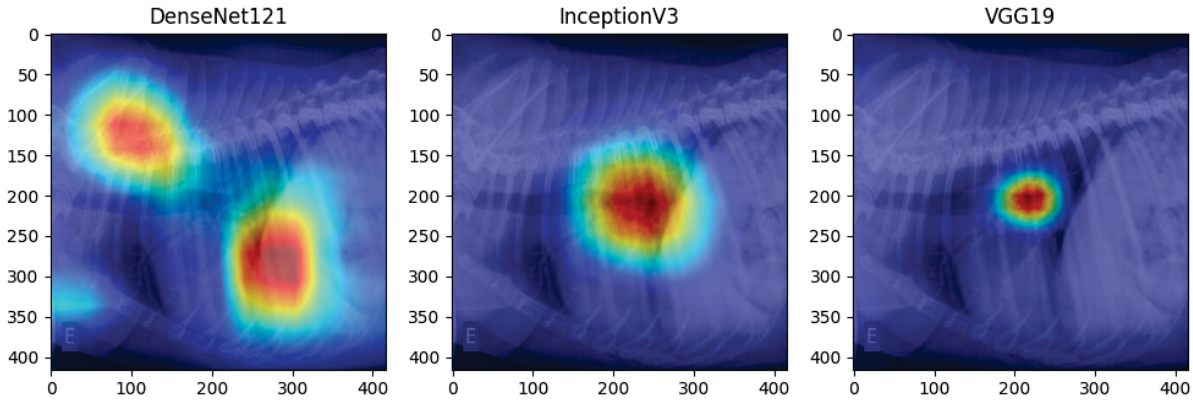


FIGURA 59 – Visualização de inferência com Grad-CAM. Classe ”aumento do átrio esquerdo” incorretamente identificada ”normal”. Fonte: O autor (2024).

Ao observar os resultados visuais gerados através da aplicação do algoritmo Grad-CAM na seção 5.5 para os pacientes classificados como ”normais”, nota-se que a DenseNet-121 tende a olhar para a imagem de maneira mais global, enquanto a InceptionV3 e principalmente a VGG19 focam na silhueta do átrio esquerdo. Já nos casos dados como ”aumento do átrio esquerdo”, as últimas camadas convolucionais dos modelos pouco ou nada são ativadas pela região do átrio esquerdo. Com isto, é possível supor que estes modelos aprenderam o contorno normal do átrio esquerdo, classificando como aumentado quando este não é encontrado na imagem.

6 CONCLUSÕES

Nesta dissertação, foi investigada a aplicação de redes neurais convolucionais (CNNs) e de vision transformers (ViTs) para a classificação do aumento do átrio esquerdo em radiografias de cães, com o objetivo de auxiliar no diagnóstico da doença mixomatosa da válvula mitral. Conforme descrito nos objetivos específicos (subseção 1.1.2), foi realizada a implementação e testes de diferentes modelos de aprendizado profundo para detecção do achado de interesse nas radiografias, foi avaliada a combinação das respostas geradas pelos modelos para a melhora no resultado obtido com um único modelo, foi aplicado o algoritmo Grad-CAM, uma técnica de inteligência artificial explicável, a qual permitiu uma compreensão visual dos modelos obtidos para análise dos resultados. Isto resultou em modelos treinados que serão disponibilizados para que outros pesquisadores os refinem com seus próprios conjuntos de dados.

Visto que não foram identificados conjuntos de dados públicos compostos por radiografias de cães com aumento do átrio esquerdo, houve necessidade da confecção de um novo *dataset*. Este, por sua vez, foi construído a partir de um levantamento realizado no banco de imagens do Hospital Veterinário da Universidade Federal do Paraná (HV-UFPR), além da colaboração de outros cinco hospitais veterinários. Desta forma, foi possível selecionar 160 pacientes com aumento do átrio esquerdo e 290 pacientes normais, totalizando 450 pacientes e 1039 imagens.

Foram selecionados sete modelos de redes neurais convolucionais (CNNs), além do vision transformer (ViT) que foi implementado também em conjunto com os mecanismos de Shifted Patch Tokenization (SPT) e Locality Self-Attention (LSA). Todos os modelos foram treinados com e sem o auxílio do aumento de dados. Utilizando a técnica de validação cruzada, foram obtidas as métricas de acurácia, sensibilidade, especificidade, precisão, F-score e AUC. Os resultados apresentados mostraram que, de maneira geral, o aumento de dados melhorou a performance das CNNs, mas o mesmo não ocorreu para o ViT. As CNNs, dada sua capacidade de se concentrarem principalmente na extração de características locais, tiveram métricas superiores ao ViT. Embora modelos baseados em ViTs tenham atingido o estado da arte em bancos de dados massivos (como JFT-300M com 300 milhões de imagens) e os mecanismos de SPT e LSA tenham mostrado bom desempenho quando testados com datasets menores como o ImageNet (com cerca de 14 milhões de imagens), o dataset utilizado neste trabalho (com pouco mais de 1 mil imagens) está numa escala de grandeza que não viu melhoria ao utilizar essas técnicas.

A combinação dos melhores modelos obtidos na forma de ensembles foi uma estratégia que se mostrou efetiva. O melhor ensemble, composto pelas redes Densenet121, VGG19 e InceptionV3, chegou a um F-score de 0.8892 ± 0.0371 e um AUC de 0.9099 ± 0.0508, superando todos os outros modelos. Os resultados obtidos neste trabalho ratificam os resultados dos trabalhos relacionados, nos quais é frequente um AUC ≥ 0.8. Contudo, não é possível fazer

um comparativo direto, uma vez que os modelos não foram treinados e testados no mesmo conjunto de dados.

As métricas alcançadas permitem afirmar que foram gerados classificadores que são de fato capazes de diferenciar entre as duas classes do problema. Sendo assim, esta é uma ferramenta capaz de colaborar com a velocidade e a assertividade do diagnóstico do aumento do átrio esquerdo nas radiografias caninas.

Embora os resultados validem a capacidade das redes neurais artificiais de classificarem o aumento do átrio esquerdo em radiografias, este trabalho apresenta algumas limitações. A primeira delas diz respeito ao tamanho do conjunto de dados (450 pacientes e 1039 imagens), uma amostra relativamente pequena para um tipo de técnica que se beneficia de grandes quantidades de dados. Além disso, como as imagens vieram de vários hospitais veterinários e de um levantamento retrospectivo, elas carecem de padronização de aquisição, seja com relação a protocolos ou equipamentos. Além disso, existe uma falta de padronização nos dados de imagem em relação a características dos animais como raça, sexo, idade e peso. Sendo assim, não é possível saber se o conjunto de dados representa fielmente a diversidade existente no mundo real.

Como trabalhos futuros estão a expansão do conjunto de dados, com a inclusão de radiografias com projeção ventrodorsal, e de radiografias que possam também apresentar outras doenças. Além disso, está a ampliação das anotações destes dados, incluindo também a segmentação das regiões e órgãos de interesse, o que estende as possibilidades de pesquisa.

Por fim, também é necessário investigar como estes modelos se comportam em um ambiente real de testes e como eles contribuem na prática com o desempenho dos veterinários no diagnóstico do aumento do átrio esquerdo em radiografias e, por consequência, da doença mixomatosa da válvula mitral.

REFERÊNCIAS

AKKAYA, Ibrahim Batuhan et al. Enhancing performance of vision transformers on small datasets through local inductive bias incorporation. **Pattern Recognition**, v. 153, p. 110510, 2024. Citado 1 vez na página 77.

BANZATO, T.; BERNARDINI, M. et al. A methodological approach for deep learning to distinguish between meningiomas and gliomas on canine MR-images. **BMC Veterinary Research**, v. 14, 2018. Citado 1 vez na página 17.

BANZATO, T.; WODZINSKI, M.; TAUCERI, F. et al. An AI-Based Algorithm for the Automatic Classification of Thoracic Radiographs in Cats. **Frontiers in Veterinary Science**, v. 8, 2021. Citado 1 vez na página 17.

BANZATO, Tommaso; WODZINSKI, Marek; BURTI, Silvia et al. Automatic classification of canine thoracic radiographs using deep learning. **Scientific Reports**, v. 11, p. 3964, 1 fev. 2021. Citado 1 vez na página 50.

BAYDAN, B.; BARIŞÇI, N.; ÜNVER, H. M. Determining the Location of Tibial Fracture of Dog and Cat Using Hybridized Mask R-CNN Architecture. In: Citado 1 vez na página 17.

BOISSADY, Emilie et al. Comparison of a Deep Learning Algorithm vs. Humans for Vertebral Heart Scale Measurements in Cats and Dogs Shows a High Degree of Agreement Among Readers. **Frontiers in Veterinary Science**, v. 8, dez. 2021. Citado 1 vez na página 50.

BUCHANAN, James W; BÜCHELER, Jörg. Vertebral scale system to measure canine heart size in radiographs. **Journal of the American Veterinary Medical Association**, Am Vet Med Assoc, v. 206, n. 2, p. 194–199, 1995. Citado 1 vez na página 51.

BURTI, S. et al. Use of deep learning to detect cardiomegaly on thoracic radiographs in dogs. **The Veterinary Journal**, v. 262, p. 105505, 2020. Citado 2 vezes nas páginas 17, 57.

CAJAL, Santiago Ramon y. **Comparative Study of the Sensory Areas of the Human Cortex**. [S.l.]: Clark University, 1899. Citado 0 vez na página 25.

CHOLLET, Francois. Xception: Deep Learning with Depthwise Separable Convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). [S.l.]: IEEE, jul. 2017. P. 1800–1807. Citado 1 vezes nas páginas 35, 37.

CHOLLET, François et al. **Keras Applications**. Disponível em: <https://keras.io/api/applications/>. Acesso em: 5 nov. 2024. Citado 0 vez na página 40.

DENG, Jia et al. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. [S.l.]: IEEE, jun. 2009. P. 248–255. Citado 1 vez na página 38.

DOSOVITSKIY, Alexey et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2021. Citado 1 vez na página 41.

ESTRADA, Amara; FOX-ALVAREZ, Stacey. Vertebral Heart Scale. **Clinician's Brief**, p. 49–53, abr. 2016. Citado 3 vezes nas páginas 21, 22.

FAN, Yu-Jiun et al. Machine Learning: Using Xception, a Deep Convolutional Neural Network Architecture, to Implement Pectus Excavatum Diagnostic Tool from Frontal-View Chest X-rays. **Biomedicines**, v. 11, p. 760, 3 mar. 2023. Citado 1 vez na página 57.

FITZKE, M. et al. RapidRead: Global Deployment of State-of-the-art Radiology AI for a Large Veterinary Teleradiology Practice. **CoRR**, abs/2111.08165, 2021. Citado 1 vez na página 17.

GÉRON, Aurélien. **Hands–On Machine Learning with Scikit–Learn and TensorFlow**. [S.l.]: O'Reilly, 2017. P. 566. Citado 1 vez na página 26.

GOMES, Daniel Adorno et al. Predicting Canine Hip Dysplasia in X-Ray Images Using Deep Learning. In: OPTIMIZATION, Learning Algorithms and Applications. Cham: Springer International Publishing, 2021. P. 393–400. Citado 1 vez na página 57.

GOOGLE. **Advanced Guide to Inception v3**. Disponível em: <https://cloud.google.com/tpu/docs/inception-v3-advanced>. Acesso em: 14 nov. 2023. Citado 0 vez na página 35.

HAHNIOSER, Richard H.R. et al. Digital selection and analogue amplification coexist in a cortex- inspired silicon circuit. **Nature**, 2000. Citado 1 vez na página 29.

HAYKIN, Simon. **Neural Networks and Learning Machines**. [S.l.: s.n.], 2008. v. 3, p. 906. Citado 2 vezes nas páginas 23, 25.

HE, Kaiming et al. Deep Residual Learning for Image Recognition, dez. 2015. Citado 1 vezes nas páginas 31–33.

_____. Identity mappings in deep residual networks. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, 9908 LNCS, 2016. Citado 1 vez na página 32.

HICKS, Steven A. et al. On evaluation metrics for medical applications of artificial intelligence. **Scientific Reports**, v. 12, p. 5979, 1 abr. 2022. Citado 1 vez na página 45.

HUANG, Jonathan et al. Speed/accuracy trade-offs for modern convolutional object detectors. In: PROCEEDINGS - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. [S.l.: s.n.], 2017. 2017-Janua, p. 3296–3305. Citado 1 vezes nas páginas 36–38.

IOFFE, Sergey; SZEGEDY, Christian. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, fev. 2015. Citado 1 vez na página 32.

KADOTANI, Saki; FRIES, Ryan. **Evaluating the Heart Size on Radiographs**. Jul. 2022. Disponível em: <https://vetmed.illinois.edu/2022/07/06/evaluating-the-heart-size-on-radiographs/#:~:text=Perform%20a%20VLAS-,1.,and%20caudal%20cardiac%20silhouette%20intersect.>. Acesso em: 6 dez. 2023. Citado 1 vez na página 21.

KEENE, Bruce W. et al. ACVIM consensus guidelines for the diagnosis and treatment of myxomatous mitral valve disease in dogs. **Journal of Veterinary Internal Medicine**, v. 33, p. 1127–1140, 3 mai. 2019. Citado 1 vez na página 16.

KHAN, Salman et al. **A Guide to Convolutional Neural Networks for Computer Vision**. 1. ed. [S.l.]: Morgan & Claypool, 2018. P. 207. Citado 3 vezes nas páginas 27–29, 46.

KUMAR, A. et al. Deep feature learning for histopathological image classification of canine mammary tumors and human breast cancer. **Information Sciences**, v. 508, p. 405–421, 2020. Citado 1 vez na página 17.

LEE, Seung Hoon; LEE, Seunghyun; SONG, Byung Cheol. **Vision Transformer for Small-Size Datasets**. [S.l.: s.n.], 2021. arXiv: 2112.13492 [cs.CV]. Citado 2 vez na página 42.

LI, Shen et al. Pilot study: Application of artificial intelligence for detecting left atrial enlargement on canine thoracic radiographs. **Veterinary Radiology & Ultrasound**, v. 61, p. 611–618, 6 nov. 2020. Citado 1 vez na página 52.

MAURÍCIO, José; DOMINGUES, Inês; BERNARDINO, Jorge. Comparing Vision Transformers and Convolutional Neural Networks for Image Classification: A Literature Review. **Applied Sciences**, v. 13, n. 9, 2023. Citado 1 vez na página 77.

MCCULLOCH, Warren S; PITTS, Walter. A Logical Calculus of Ideas Immanent in Nervous Activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 127–147, 1943. Citado 1 vez na página 23.

METZ, Charles E. Basic principles of ROC analysis. **Seminars in Nuclear Medicine**, v. 8, p. 283–298, 4 out. 1978. Citado 1 vez na página 48.

MUJAHID, Muhammad et al. Pneumonia Classification from X-ray Images with Inception-V3 and Convolutional Neural Network. **Diagnostics**, v. 12, n. 5, 2022. Citado 1 vez na página 57.

NAM, Ju Gang et al. Automatic prediction of left cardiac chamber enlargement from chest radiographs using convolutional neural network. **European Radiology**, v. 31, p. 8130–8140, 11 nov. 2021. Citado 1 vez na página 50.

O'BRIEN, M. J.; BEIJERINK, N. J.; WADE, C. M. Genetics of canine myxomatous mitral valve disease. **Animal Genetics**, v. 52, p. 409–421, 4 ago. 2021. Citado 2 vezes nas páginas 16, 19.

OH, Jun-Young et al. Leveraging Image Classification and Semantic Segmentation for Robust Cardiomegaly Diagnosis in Pet. **The Journal of Korean Institute of Information Technology**, v. 21, p. 143–152, 8 ago. 2023. Citado 1 vez na página 52.

OZDEMIR, Sinan. **Principles of Data Science**. [S.l.]: Packt Publishing, 2017. P. 388. Citado 1 vez na página 44.

PARKER, Heidi G.; KILROY-GLYNN, Paul. Myxomatous mitral valve disease in dogs: Does size matter? **Journal of Veterinary Cardiology**, v. 14, p. 19–29, 1 mar. 2012. Citado 1 vez na página 16.

PATTARAMANEE, A. et al. Computer-Aided Diagnosis for Lung Lesion in Companion Animals from X-ray Images Using Deep Learning Techniques. In: p. 1–6. Citado 2 vez na página 17.

RESTANI, Guilherme M. **Identificação de faltas em sistemas de distribuição de energia elétrica com aprendizado profundo e processamento de imagens**. 2019. F. 121. Monografia (Bacharelado em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Universidade Federal do Paraná - UFPR, Curitiba. Citado 0 vezes nas páginas 24, 26, 45, 47.

ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological Review**, v. 65, n. 6, p. 386–408, 1958. Citado 1 vez na página 23.

RUSSAKOVSKY, Olga et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015. Citado 1 vez na página 31.

SAJID, Muhammad Zaheer et al. DR-NASNet: Automated System to Detect and Classify Diabetic Retinopathy Severity Using Improved Pretrained NASNet Model. **Diagnostics**, v. 13, p. 2645, 16 ago. 2023. Citado 1 vez na página 57.

SAXENA, Ashendra Kumar; SASTRY, Divakara E.V.; ROOPASHREE. Canine Thoracic Radiographs Classification Using Deep Learning Algorithms: An Investigation. **Revista Electronica de Veterinaria**, v. 24, 2 2023. Citado 2 vezes nas páginas 51, 57.

SELVARAJU, Ramprasaath R. et al. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. **International Journal of Computer Vision**, v. 128, p. 336–359, 2 fev. 2020. Citado 2 vezes nas páginas 57, 83.

SIMONYAN, K; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: p. 1–14. Citado 1 vez na página 30.

SOLOMON, J. et al. Diagnostic validation of vertebral heart score machine learning algorithm for canine lateral chest radiographs. **Journal of Small Animal Practice**, ago. 2023. Citado 1 vez na página 51.

SUN, Chen et al. Revisiting Unreasonable Effectiveness of Data in Deep Learning Era. In: 2017 IEEE International Conference on Computer Vision (ICCV). Los Alamitos, CA, USA: IEEE Computer Society, out. 2017. P. 843–852. Citado 1 vez na página 42.

SZEGEDY, Christian; LIU, Wei et al. Going Deeper with Convolutions, set. 2014. Citado 2 vezes nas páginas 33, 34.

SZEGEDY, Christian; VANHOUCKE, Vincent et al. Rethinking the Inception Architecture for Computer Vision. **Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition**, 2016-December, 2016. Citado 3 vezes nas páginas 33–35.

TILLEY, Larry P. et al. **Manual of Canine and Feline Cardiology**. 4th. Missouri: Saunders Elsevier, 2008. Citado 6 vezes nas páginas 16, 19, 20.

VALENTE, Carlotta et al. Development of an artificial intelligence-based method for the diagnosis of the severity of myxomatous mitral valve disease from canine chest radiographs. **Frontiers in Veterinary Science**, v. 10, set. 2023. Citado 1 vez na página 51.

VASWANI, Ashish et al. Attention is all you need. In: PROCEEDINGS of the 31st International Conference on Neural Information Processing Systems. Long Beach, California, USA: Curran Associates Inc., 2017. (NIPS'17), p. 6000–6010. Citado 1 vez na página 41.

YOON, Y.; HWANG, T.; LEE, H. Prediction of radiographic abnormalities by the use of bag-of-features and convolutional neural networks. **The Veterinary Journal**, v. 237, p. 43–48, jul. 2018. Citado 1 vez na página 50.

ZOPH, Barret; LE, Quoc. Neural Architecture Search with Reinforcement Learning. In: INTERNATIONAL Conference on Learning Representations. [S.l.: s.n.], 2017. Citado 1 vez na página 38.

ZOPH, Barret; VASUDEVAN, Vijay et al. Learning Transferable Architectures for Scalable Image Recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. [S.l.]: IEEE, jun. 2018. P. 8697–8710. Citado 1 vezes nas páginas 38, 40.