



Thiago Henrique Leite da Silva

Análise comparativa de redes neurais convolucionais na classificação de tumores pulmonares

São José dos Campos, SP

Thiago Henrique Leite da Silva

Análise comparativa de redes neurais convolucionais na classificação de tumores pulmonares

Trabalho de conclusão de curso apresentado ao
Instituto de Ciência e Tecnologia – UNIFESP,
como parte das atividades para obtenção do tí-
tulo de Bacharel em Ciência da Computação.

Universidade Federal de São Paulo – UNIFESP

Instituto de Ciência e Tecnologia

Bacharelado em Ciência da Computação

Orientador: Prof^a. Dra. Lilian Berton

São José dos Campos, SP

Julho de 2023

Thiago Henrique Leite da Silva

Análise comparativa de redes neurais convolucionais na classificação de tumores pulmonares

Trabalho de conclusão de curso apresentado ao
Instituto de Ciência e Tecnologia – UNIFESP,
como parte das atividades para obtenção do tí-
tulo de Bacharel em Ciência da Computação.

Trabalho aprovado em 12 de Julho de 2023:

Prof^a. Dra. Lilian Berton
Orientador

Prof. Dr. Luis Augusto Martins Pereira
Convidado 1

Prof^a. Dra. Thaína Aparecida Azevedo Tosta
Convidado 2

São José dos Campos, SP
Julho de 2023

*Este trabalho é dedicado aos meus pais e meus irmãos, que foram minha base para que
chegasse até aqui.*

Agradecimentos

Agradeço primeiramente a Deus, que em meio as inúmeras dificuldades enfrentadas durante o curso, jamais me desamparou e colocou pessoas especiais em minha vida que foram fundamentais durante esta jornada.

À meus familiares, em especial meus pais, Alcione Eugênia e Renato Leite, e meus irmãos, Samuel Leite e Thalita Beatriz, que me apoiaram incondicionalmente durante toda a minha vida.

À minha tia Elisabeth Eugênia (in memoriam), que foi uma das inspirações para a escolha do tema abordado no trabalho.

À todos os meus professores, mestres e doutores com quem tive o prazer de dividir a sala de aula durante a escola e universidade, sem eles nada disso seria possível.

À Prof^a. Dr^a. Lilian Berton, por ter aceitado me orientar durante o trabalho, fazendo isso com extremo zelo e excelência.

À todos aqueles que fizeram parte da minha vida em algum momento, quem sou hoje é resultado da soma de todas as experiências e trocas que pude ter com cada um.

*“Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito.
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

O câncer de pulmão é uma das principais causas de mortes no mundo atualmente, e na grande maioria dos casos, o diagnóstico da doença ocorre quando o paciente já se encontra em estágios avançados da mesma e com baixa expectativa de cura. Através do aprendizado de máquina por redes neurais convolucionais, unido ao processamento de imagens digitais, podemos desenvolver algoritmos capazes de auxiliar profissionais de saúde no diagnóstico do câncer de pulmão. Este trabalho tem por objetivo, portanto, realizar a análise comparativa de diferentes redes neurais convolucionais atuando neste processo. Foram comparadas quatro redes neurais convolucionais clássicas (VGG16, VGG19, Resnet e Xception) em duas bases de dados, sendo uma binária e outra multiclasse (com três classes). As duas redes com os melhores desempenhos foram Xception e Resnet, respectivamente, ambas obtendo 99% de acurácia nos conjuntos de dados binário e multiclasse. Concluimos que as redes neurais convolucionais clássicas conseguem ser eficientes para as duas abordagens. Por meio da análise dos mapas de calor, os quais fornecem uma representação visual das áreas de uma imagem que mais contribuem para a resposta ou classificação dada pela rede neural, notou-se que algumas áreas do pulmão foram mais relevantes para a classificação final.

Palavras-chaves: redes neurais convolucionais. câncer de pulmão. processamento de imagens.

Abstract

Lung cancer is one of the main causes of death in the world today, and in the vast majority of cases, the diagnosis of the disease occurs when the patient is already in advanced stages of the same and with low expectation of cure. Through machine learning through convolutional neural networks, together with digital image processing, we can develop algorithms capable of assisting health professionals in the diagnosis of lung cancer. This work aims, therefore, to carry out a comparative analysis of different convolutional neural networks acting in this process. Four classic convolutional neural networks (VGG16, VGG19, Resnet and Xception) were compared in two databases, one binary and the other multiclass (with three classes). The two best performing networks were Xception and Resnet, respectively, both achieving 99% accuracy on the binary and multiclass datasets. We conclude that classical convolutional neural networks can be efficient for both approaches. Through the analysis of heat maps, which provide a visual representation of the areas of an image that most contribute to the response or classification given by the neural network, it was noted that some lung areas were more relevant to the final classification.

Key-words: convolutional neural networks. lung cancer. image processing.

Lista de ilustrações

Figura 1 – Número estimado de casos de câncer em 2020 no mundo.	26
Figura 2 – Número estimado de casos de câncer em 2020 no mundo por tipo de câncer. .	27
Figura 3 – Tipos de Aprendizado de Máquina.	31
Figura 4 – Etapas para aplicação de técnicas de AM.	33
Figura 5 – Neurônio Biológico.	35
Figura 6 – Neurônio Artificial.	37
Figura 7 – Exemplo de gráfico de Função com Limite.	38
Figura 8 – Exemplo de gráfico de Função Sigmoides.	39
Figura 9 – Exemplo de um modelo de aprendizado profundo.	41
Figura 10 – Exemplo de um modelo de CNN.	41
Figura 11 – Demonstração da camada de Convolução.	42
Figura 12 – Demonstração da camada de Pooling.	44
Figura 13 – Comparativo de redes do tipo VGG.	45
Figura 14 – Arquitetura das redes VGG19 e Resnet.	46
Figura 15 – Arquitetura da rede Xception.	47
Figura 16 – Representação de Imagem Digital.	48
Figura 17 – Representação da Vizinhança-4 de um Pixel.	49
Figura 18 – Representação da Vizinhança-8 de um Pixel.	50
Figura 19 – Exemplos de imagens do conjunto de dados SPIE-AAPM-NCI.	
0 - Malignos e 1 - Benignos	60
Figura 20 – Exemplos de imagens do conjunto de dados IQ-OTH/NCCD.	
0 - Benigno, 1 - Maligno e 2 - Saudável	61
Figura 21 – Exemplo de imagem redimensionada com 180x180 pixels.	63
Figura 22 – Processo de aumento de dados.	65
Figura 23 – <i>Dataset</i> Binário - Modelo VGG16.	68
Figura 24 – <i>Dataset</i> Binário - Resultados VGG16.	69
Figura 25 – <i>Dataset</i> Binário - Modelo VGG19.	70
Figura 26 – <i>Dataset</i> Binário - Resultados VGG19.	71
Figura 27 – <i>Dataset</i> Binário - Modelo Resnet.	72
Figura 28 – <i>Dataset</i> Binário - Resultados Resnet.	73
Figura 29 – <i>Dataset</i> Binário - Modelo Xception.	74
Figura 30 – <i>Dataset</i> Binário - Resultados Xception.	75
Figura 31 – <i>Dataset</i> Binário - VGG16 - Mapa de calor para tumor benigno.	76

Figura 32 – <i>Dataset</i> Binário - VGG16 - Mapa de calor para tumor maligno.	76
Figura 33 – <i>Dataset</i> Binário - VGG19 - Mapa de calor para tumor benigno.	77
Figura 34 – <i>Dataset</i> Binário - VGG19 - Mapa de calor para tumor maligno.	77
Figura 35 – <i>Dataset</i> Binário - Resnet - Mapa de calor para tumor benigno.	77
Figura 36 – <i>Dataset</i> Binário - Resnet - Mapa de calor para tumor maligno.	77
Figura 37 – <i>Dataset</i> Binário - Xception - Mapa de calor para tumor benigno.	78
Figura 38 – <i>Dataset</i> Binário - Xception - Mapa de calor para tumor benigno.	78
Figura 39 – <i>Dataset</i> Binário - Matriz de Confusão VGG16.	79
Figura 40 – <i>Dataset</i> Binário - Matriz de Confusão VGG19.	80
Figura 41 – <i>Dataset</i> Binário - Matriz de Confusão Resnet.	80
Figura 42 – <i>Dataset</i> Binário - Matriz de Confusão Xception.	81
Figura 43 – Exemplo de imagem redimensionada com 224x224 pixels.	83
Figura 44 – <i>Dataset</i> Multiclasse - Modelo Resnet.	85
Figura 45 – <i>Dataset</i> Multiclasse - Resultados Resnet.	86
Figura 46 – <i>Dataset</i> Multiclasse - Modelo Xception.	87
Figura 47 – <i>Dataset</i> Multiclasse - Resultados Xception.	88
Figura 48 – <i>Dataset</i> Multiclasse - Resnet - Mapa de calor para tumor benigno.	89
Figura 49 – <i>Dataset</i> Multiclasse - Resnet - Mapa de calor para tumor maligno.	89
Figura 50 – <i>Dataset</i> Multiclasse - Resnet - Mapa de calor para pulmões sem tumor.	89
Figura 51 – <i>Dataset</i> Multiclasse - Xception - Mapa de calor para tumor benigno.	90
Figura 52 – <i>Dataset</i> Multiclasse - Xception - Mapa de calor para tumor maligno.	90
Figura 53 – <i>Dataset</i> Multiclasse - Xception - Mapa de calor para pulmões sem tumor.	90
Figura 54 – <i>Dataset</i> Multiclasse - Matriz de Confusão Resnet.	91
Figura 55 – <i>Dataset</i> Multiclasse - Matriz de Confusão Xception.	91

Lista de tabelas

Tabela 1 – Exemplo de Matriz de Confusão	52
Tabela 2 – Resumo da Revisão Bibliográfica	56
Tabela 3 – Detalhes do conjunto de dados SPIE-AAPM-NCI	59
Tabela 4 – <i>Dataset</i> Binário - Resumo dos resultados do modelo VGG16	66
Tabela 5 – <i>Dataset</i> Binário - Resumo dos resultados do modelo VGG19	66
Tabela 6 – <i>Dataset</i> Binário - Resumo dos resultados do modelo Resnet	67
Tabela 7 – <i>Dataset</i> Binário - Resumo dos resultados do modelo Xception	67
Tabela 8 – Quantidade de parâmetros dos modelos treinados	79
Tabela 9 – <i>Dataset</i> Multiclasse - Resumo dos resultados do modelo Resnet	84
Tabela 10 – <i>Dataset</i> Multiclasse - Resumo dos resultados do modelo Xception	84

Lista de abreviaturas e siglas

INCA	Instituto Nacional do Câncer
DNA	Ácido Desoxirribonucleico
OMS	Organização Mundial da Saúde
CDC	Centro de Controle e Prevenção de Doenças
SVM	<i>Support Vector Machine</i>
ANN	<i>Artificial Neural Networks</i>
CNN	<i>Convolutional Neural Networks</i>
MLP	<i>Multilayer Perceptron</i>
KNN	<i>K-Nearest Neighbors</i>
VGG	<i>Very Deep Convolutional Networks</i>
Resnet	<i>Residual Neural Network</i>
Xception	<i>Extreme Inception</i>
AM	Aprendizado de Máquina
IA	Inteligência Artificial
TC	Tomografia Computadorizada
DICOM	<i>Digital Imagins and Communications in Medicine</i>
IBM	<i>International Business Machines Corporation</i>
TP	<i>True Positive</i>
TN	<i>True Negative</i>
FP	<i>False Positive</i>
FN	<i>False Negative</i>
prel	<i>positive reliability</i>
nrel	<i>negative reliability</i>

sup	<i>support</i>
sens	<i>sensitivity</i>
spec	<i>specificity</i>
tacc	<i>total accuracy</i>
cov	<i>coverage</i>
DTCNN	<i>Deep Transfer Convolutional Neural Network</i>
ELM	<i>Extreme Learning Machine</i>
LIDC-IDRI	<i>Lung Image Database Consortium - Image Database Resource Initiative</i>
FAH-GMU	<i>First Affiliated Hospital of Guangzhou Medical University in China</i>
XGBoost	<i>Extreme Gradient Boosting</i>
TPE	<i>Tree Parzen Estimator</i>
DT	<i>Decision Tree</i>
3D	Tridimensional
2D	Bidimensional
AAPM	<i>American Association of Physicists in Medicine</i>
NCI	<i>National Cancer Institute</i>
Grad-CAM	<i>Gradient-weighted Class Activation Mapping</i>
RM	Ressonâncias Magnéticas

Lista de símbolos

\in	pertence
φ	letra grega Fi
Σ	sigma representando a função de Somatório
$\%$	porcentagem
\forall	para todo
α	alpha
β	beta
γ	gamma
δ	delta
λ	lambda
\cdot	ponto de multiplicação (ponto centralizado)
\times	cruz de multiplicação
\neq	símbolo de não igual
\in	símbolo "contido em"
\leq	símbolo de menor ou igual
\geq	símbolo de maior ou igual
$+$	símbolo de soma
$-$	símbolo de subtração
$/$	símbolo de divisão
f	função

Sumário

1	Introdução	25
1.1	Contextualização e motivação	25
1.2	Definição do problema	27
1.3	Justificativas	28
1.4	Objetivos	29
1.4.1	Objetivo Geral	29
1.4.2	Objetivo Específico	29
1.5	Organização do documento	29
2	Fundamentação Teórica	31
2.1	Aprendizado de Máquina	31
2.1.1	Paradigmas de Aprendizado de Máquina	33
2.1.1.1	Baseado em Exemplos	34
2.1.1.2	Conexionista	34
2.1.1.3	Estatístico	34
2.1.1.4	Genético	34
2.1.1.5	Simbólico	34
2.2	Redes Neurais	34
2.2.1	Neurônio Biológico	35
2.2.2	Neurônio Artificial	36
2.2.3	Funções de Ativação	38
2.2.3.1	Função com Limite (<i>Threshold</i>)	38
2.2.3.2	Função Sigmoides	38
2.3	Aprendizado Profundo	39
2.3.1	Redes Neurais Convolucionais	40
2.3.1.1	Convolução	41
2.3.1.2	<i>Pooling</i>	43
2.3.1.3	VGG	43
2.3.1.4	Resnet	44
2.3.1.5	Xception	45
2.4	Processamento de Imagens	47
2.4.1	Representação Digital de Imagens	48
2.4.2	Pixels	49
2.4.3	Etapas do Processamento de Imagens	50
2.5	Avaliação de Algoritmos	51

2.5.1	Taxas de Acurácia e Erro	51
2.5.2	Matriz de Confusão	51
2.5.3	<i>Gradient-weighted Class Activation Mapping</i>	53
3	Revisão Bibliográfica	55
3.1	Trabalhos Relacionados	55
3.2	Resumo	56
3.3	Conclusões	57
4	Proposta de Trabalho	59
4.1	Bases de Dados	59
4.1.1	<i>SPIE-AAPM-NCI Lung Nodule Classification</i>	59
4.1.2	<i>IQ-OTH/NCCD</i>	60
4.2	Softwares e Bibliotecas	61
4.3	Método e Análise da Proposta	62
5	Resultados	63
5.1	<i>Dataset</i> Binário: SPIE-AAPM-NCI	63
5.1.1	Tratamento das Imagens	63
5.1.2	Execução dos Modelos	64
5.1.2.1	VGG16	66
5.1.2.2	VGG19	66
5.1.2.3	Resnet	66
5.1.2.4	Xception	67
5.1.3	Grad-CAM	76
5.1.3.1	VGG16	76
5.1.3.2	VGG19	76
5.1.3.3	Resnet	77
5.1.3.4	Xception	77
5.1.4	Parâmetros das Redes	78
5.1.5	Matriz de Confusão	79
5.1.5.1	VGG16	79
5.1.5.2	VGG19	80
5.1.5.3	Resnet	80
5.1.5.4	Xception	81
5.1.6	Avaliação dos Resultados	82
5.2	<i>Dataset</i> Multiclasse: IQ-OTH/NCCD	82
5.2.1	Tratamento das Imagens	82
5.2.2	Execução dos Modelos	83

5.2.2.1	Resnet	84
5.2.2.2	Xception	84
5.2.3	Grad-CAM	88
5.2.3.1	Resnet	89
5.2.3.2	Xception	90
5.2.4	Matriz de Confusão	91
5.2.4.1	Resnet	91
5.2.4.2	Xception	91
5.2.5	Avaliação dos Resultados	92
6	Considerações Finais	93
	Referências	95

1 Introdução

Neste capítulo são apresentadas a contextualização do tema, a definição do problema e os objetivos do trabalho.

1.1 Contextualização e motivação

O termo câncer engloba dezenas de doenças com dois atributos principais em comum: a multiplicação desordenada de células malignas, que serão explicadas mais adiante, e a capacidade destas células invadirem tecidos adjacentes ou órgãos à distância através da corrente sanguínea e do sistema linfático, ação denominada metástase. A doença se origina por conta de uma mutação no ácido desoxirribonucleico (DNA) de uma célula, que passará a exercer suas atividades de forma errônea. Quando ocorre o acúmulo de células com essas características em um determinado órgão, seu pleno funcionamento é comprometido. Todo o processo de evolução e desenvolvimento do câncer é denominado carcinogênese, um processo que, em linhas gerais, pode levar anos para ocorrer por completo e começar a exibir os primeiros indícios visíveis da doença ao paciente ([INCA, 2020](#)).

São três os estágios principais da carcinogênese: iniciação, promoção e progressão. Na iniciação é onde as células sofrerão alguma mutação em seu DNA por conta dos agentes cancerígenos, substâncias capazes de alterar o código genético de uma célula. Já na promoção, de forma vagarosa a célula iniciada começa a se tornar maligna, porém os efeitos dessa alteração não podem ainda ser sentidos ou vistos com facilidade. Esse estágio é muito importante no processo de desenvolvimento da doença, pois a eliminação do contato com os agentes promotores do câncer pode impedir com que a célula continue sua evolução para se tornar uma célula maligna. No terceiro e último estágio, o de progressão, as células que se tornaram malignas na etapa anterior já estão se multiplicando de forma descontrolada e irreversível, é apenas uma questão de tempo para as primeiras manifestações clínicas do câncer no paciente. A questão fundamental nesta etapa é se atentar aos primeiros indícios dessa manifestação para o rápido diagnóstico da doença ([INCA, 2021](#)).

Por ser um problema que se origina em uma célula que sofreu uma mutação, o câncer é algo que pode ocorrer em qualquer parte do corpo, no entanto, algumas partes são mais afetadas que outras e dependendo do tipo da célula que sofreu a mutação o câncer pode ser mais ou menos agressivo. Neste trabalho, a palavra câncer será utilizada de forma a englobar todos os tipos possíveis da doença.

O câncer é bastante conhecido e temido por sua tamanha letalidade. Em 2020, a Organiza-

ção Mundial da Saúde (OMS) estimou a incidência de pouco mais de 19 milhões de novos casos de câncer, como mostrado na Figura 1. Sendo 52,17% em homens e 47,83% em mulheres, com uma taxa de mortalidade de 51,61%, o que equivale a quase 10 milhões de pessoas que perderam suas vidas por conta da doença no ano em questão. Das vítimas fatais, mais de 88,19% das pessoas possuíam mais de 50 anos, o que evidencia o fato da agressividade da doença ser diretamente proporcional ao avanço da idade do paciente.

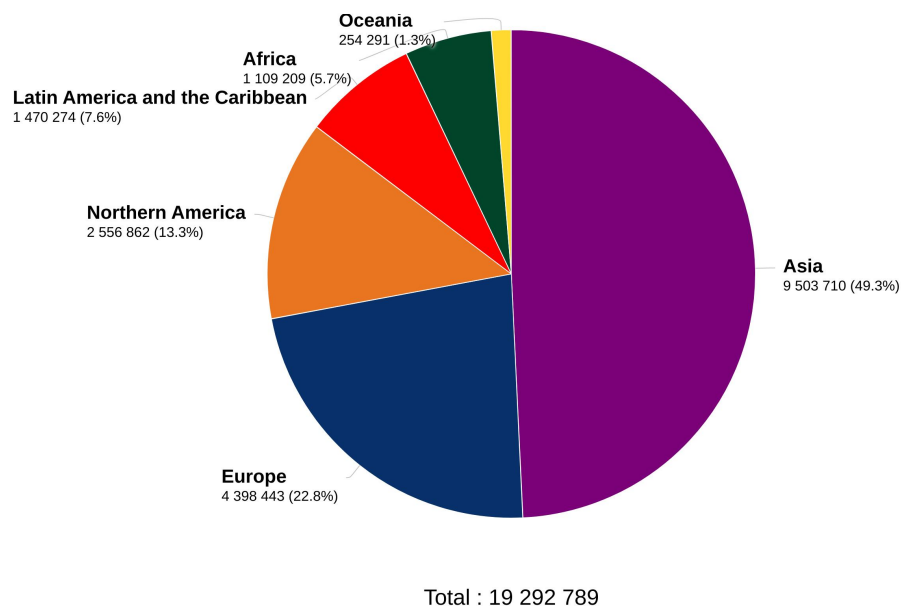


Figura 1 – Número estimado de casos de câncer em 2020 no mundo.

Fonte: (OMS, 2021)

Tratando-se agora da incidência por tipo de câncer, como podemos ver na Figura 2, os três tipos com maior incidência foram: câncer de mama (11,7%), câncer de pulmão (11,4%) e câncer de próstata (7,3%). Quando observamos a mortalidade de acordo com a região do câncer, podemos começar a diferenciar quais delas são mais críticas quando afetadas; as três maiores taxas de mortalidade foram: câncer de pulmão (18%), câncer de fígado (8,3%) e câncer de estômago (7,7%).

O Centro de Controle e Prevenção de Doenças (CDC, 2021) registrou que em 2020, nos Estados Unidos, o câncer foi a segunda maior causa de mortes no país, ficando atrás apenas de mortes relacionadas a doenças cardíacas. Referindo-se agora ao Brasil, foram registradas mais de 225 mil mortes por câncer nesse mesmo ano, sendo a principal causa o câncer de pulmão, traqueia e brônquios para os homens (13,6%), e para as mulheres o câncer de mama (16,5%). A nível mundial o câncer de pulmão aparece na sexta colocação do ranking das 10 principais doenças que mais resultaram em mortes no mundo no ano de 2019 (OMS, 2021). Além disso, aproximadamente 70% dos casos de morte ocorrem em países de baixa ou média renda, e 30% deles poderiam ter sido curados se detectados precocemente para o início do tratamento (INCA, 2020).

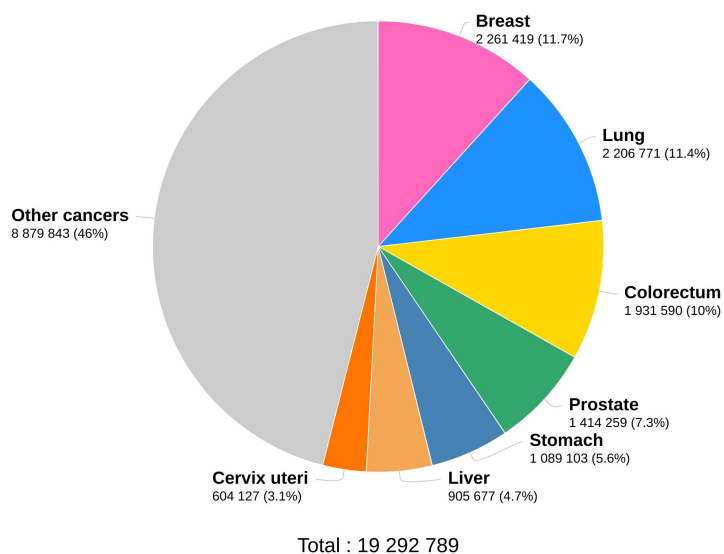


Figura 2 – Número estimado de casos de câncer em 2020 no mundo por tipo de câncer.

Fonte: (OMS, 2021)

Cerca de 27% da população Brasileira faz uso do sistema de saúde privado, e os 73% restantes do sistema público, segundo dados do Instituto Brasileiro de Geografia e Estatística (IBGE, 2008), diferença que impacta diretamente nos índices de diagnóstico da doença, eficácia do tratamento e expectativa de cura. Um fato que muitas pessoas desconhecem é que, apesar da questão genética afetar no desenvolvimento de um câncer, esta não é a principal causalidade da doença, e sim fatores naturais causados pelo próprio ser humano, como exposição excessiva ao sol, uma das principais causas do câncer de pele; o tabagismo, um dos principais causadores do câncer de pulmão; entre outras ações que podem fazer com que uma de nossas células sofram uma mutação e se tornem malignas (INCA, 2020).

Como podemos observar analisando os números vistos relacionados a incidência e mortes por câncer no mundo, um dos órgãos onde a doença acaba sendo mais agressiva é o pulmão, por impactar diretamente as vias respiratórias. O mais interessante é que o câncer de pulmão é, conforme classificado pelo INCA, uma das principais causas de morte evitáveis no mundo, já que a grande maioria dos casos, aproximadamente 85%, foram desencadeados pelo tabagismo ou pela exposição passiva ao tabaco. Pelo fato desse tipo de câncer ser uma das principais causas de morte relacionadas à doença, este trabalho visa colaborar de alguma forma no reconhecimento automático do câncer de pulmão em tomografias computadorizadas (INCA, 2020).

1.2 Definição do problema

O diagnóstico do câncer de pulmão tardio é um grave problema enfrentado atualmente no Brasil, onde as pessoas, na grande maioria dos casos, são diagnosticadas em estágios avançados da doença, quando a chance de cura é muito baixa ou inexistente do ponto de vista da medicina.

De acordo com a *Cancer Research UK* ([RESEARCH, 2018](#)), dos pouco mais de 20 mil pacientes com câncer que eles possuem cadastrados em seu sistema, apenas 8,85% tiveram a doença diagnosticada no primeiro estágio.

Sendo assim, é de extrema necessidade buscarmos formas de otimizar o diagnóstico do câncer de pulmão na população de nosso país. Existem centenas de pessoas doentes que ainda não foram diagnosticadas, e o simples fato de um diagnóstico precoce pode levá-las a tomar as devidas intervenções necessárias para salvar sua vida. Como forma de democratizar o acesso à informação e reduzir o impacto que as diferenças sociais causam na taxa de diagnóstico precoce da doença, o objetivo deste trabalho é avaliar e definir qual modelo de rede neural convolucional melhor conseguiria auxiliar um profissional da saúde no diagnóstico da doença.

Diversas técnicas relacionadas ao aprendizado de máquina têm sido utilizadas na análise de imagens torácicas e no auxílio da detecção de câncer. [Chaturvedi et al. \(2021\)](#), elaborou um resumo dos resultados obtidos nos principais trabalhos realizados por pesquisadores da área de aprendizado de máquina aplicadas ao diagnóstico do câncer de pulmão, algumas destas técnicas empregadas nos trabalhos analisados foram *Support Vector Machine* (SVM) e *Artificial Neural Networks* (ANN), que geraram excelentes resultados com altas acurácias.

1.3 Justificativas

Alguns trabalhos da literatura já exploraram a detecção de câncer de pulmão. Por exemplo, [Rodrigues et al. \(2018\)](#), compararam a utilização de três diferentes técnicas em seu trabalho: *Multilayer Perceptron* (MLP), SVM e *K-Nearest Neighbor* (KNN). Os resultados obtidos foram uma acurácia de 95,40%, 96,70% e 95,30%, respectivamente. Já [Nasser e Abu-Naser \(2019\)](#) abordou o tema aplicando ANNs, obtendo uma acurácia de 96,67%.

Pelos excelentes resultados já obtidos por outros pesquisadores do assunto, sabemos que a aplicação de redes neurais a esta problemática traz soluções promissoras. Neste trabalho, pretende-se comparar especificamente modelos clássicos de CNNs, tais como a VGG desenvolvida em 2014, que é caracterizada por ter arquiteturas profundas com um número maior de camadas convolucionais. As versões mais conhecidas são VGG16 e VGG19, que possuem 16 e 19 camadas, respectivamente ([SIMONYAN; ZISSERMAN, 2014](#)).

A Resnet apresentada pela *Microsoft Research* em 2015, é conhecida por suas conexões de "salto" (*skip connections*) que permitem a passagem direta de informações de uma camada para outra, ajudando a aliviar o problema de desvanecimento do gradiente em redes mais profundas ([HE et al., 2016](#)).

A rede Xception (*Extreme Inception*) também é uma arquitetura de rede neural convolucional que foi proposta por François Chollet em 2016. É uma variante do modelo *Inception*, que visa melhorar ainda mais o desempenho em termos de eficiência e precisão, por meio de conexões

residuais, que permitem que os dados fluam diretamente entre as camadas, pulando a convolução. Essa técnica ajuda a resolver o problema de desvanecimento do gradiente (*vanishing gradient*) e acelera o treinamento (CHOLLET, 2017). Todas estas redes estão implementadas em bibliotecas como Keras e Tensorflow, sendo de fácil uso e manipulação.

É de suma importância realizar comparações entre redes neurais convolucionais a fim de determinar qual delas seria mais eficaz no auxílio ao diagnóstico do câncer de pulmão, priorizando aquela com uma taxa de assertividade mais elevada na classificação dos tumores. No contexto de comparar redes que compartilham a mesma arquitetura, como as redes VGG16 e VGG19, é relevante analisar os critérios que podem influenciar na obtenção de resultados superiores dentro da mesma família de redes. Além disso, a utilização de mapas de calor é empregada para auxiliar na interpretação das decisões da rede neural e aprimorar o desempenho do modelo por meio da depuração.

1.4 Objetivos

1.4.1 Objetivo Geral

Analisar o desempenho de redes neurais convolucionais clássicas na classificação de tumores pulmonares, com base em imagens de tomografia computadorizada do pulmão, a fim de identificar o modelo mais eficiente para auxiliar profissionais de saúde no diagnóstico da doença.

1.4.2 Objetivo Específico

- Analisar os desafios enfrentados no diagnóstico do câncer de pulmão e explorar a diversidade de tipos de tumores pulmonares;
- Explorar e destacar as vantagens e benefícios da utilização de redes neurais convolucionais na classificação de imagens para o diagnóstico do câncer de pulmão;
- Realizar uma análise comparativa do desempenho de diferentes arquiteturas de redes neurais convolucionais no processamento de imagens de tumores pulmonares, considerando métricas como acurácia, sensibilidade e precisão;
- Aplicar mapas de calor para identificar as regiões mais relevantes na imagem para as redes neurais convolucionais.

1.5 Organização do documento

No Capítulo 2, de fundamentação teórica, foram abordados conceitos essenciais relacionados ao tema do trabalho. Tópicos como aprendizado de máquina, processamento de imagens,

redes neurais e, em particular, redes neurais convolucionais. Serão apresentados exemplos relevantes de redes neurais convolucionais, como VGG16, VGG19, Resnet e Xception, destacando suas arquiteturas e aplicações.

No Capítulo 3, de revisão bibliográfica, foi realizada uma análise crítica de trabalhos anteriores sobre o mesmo tema. Foram revisados estudos e pesquisas relacionados ao uso de redes neurais convolucionais no diagnóstico do câncer de pulmão, com o objetivo de compreender os resultados alcançados e identificar as lacunas existentes. Essa revisão bibliográfica permitirá identificar as contribuições adicionais que podem ser oferecidas por este trabalho.

No Capítulo 4, de proposta do trabalho, foram apresentados detalhes sobre os softwares utilizados na implementação deste estudo, o conjunto de dados utilizado e a metodologia adotada para a avaliação dos resultados. Também foram descritos os critérios de seleção do conjunto de dados e as etapas de preparação das imagens para treinamento e teste dos modelos de redes neurais convolucionais. Além disso, foram apresentados os passos e etapas do experimento, incluindo a configuração dos hiperparâmetros dos modelos e as métricas de avaliação utilizadas.

No Capítulo 5, de resultados, foram apresentadas as análises e comparações do desempenho de cada uma das redes neurais convolucionais utilizadas neste trabalho. Além disso, foram discutidos os resultados obtidos, considerando métricas de avaliação relevantes, como acurácia e precisão.

Por fim, no Capítulo 6, de considerações finais, foram apresentadas as principais observações e conclusões obtidas com base nos resultados alcançados. Esta seção fornecerá uma visão geral das descobertas mais significativas e suas implicações, consolidando o conhecimento adquirido ao longo deste trabalho.

2 Fundamentação Teórica

2.1 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um ramo da área de Inteligência Artificial (IA) que tem por finalidade desenvolver sistemas que possuem a capacidade de aprender a construir novos conhecimentos de maneira automatizada a partir de dados de treinamento. O processo de aprendizagem de um sistema é caracterizado pelo aprendizado indutivo, que pode ser não supervisionado ou supervisionado, este último por sua vez é utilizado para classificação e regressão. No aprendizado supervisionado, fornecemos ao algoritmo um conjunto de dados já classificados, para que, a partir do reconhecimento de padrões, ele consiga construir um classificador capaz de nos informar, para exemplos ainda não classificados, qual a sua classe. Quando os valores dessas classes são discretos, descrevemos o problema como classificação, caso estejamos trabalhando com valores contínuos, descrevemos o problema como regressão (MONARD; BARANAUSKAS, 2003). Em relação ao aprendizado não supervisionado, o algoritmo receberá dados que não estão classificados, sendo assim, seu objetivo é verificar se os dados recebidos podem ser agrupados de alguma forma, gerando esses agrupamentos, se possível. Feito isso, é iniciado o processo de análise dos grupos formados para se definir o que cada um deles significa, inseridos no contexto da problemática (HANSON; STUTZ; CHEESEMAN, 1991). A Figura 3 exibe os tipos de aprendizados de máquina.

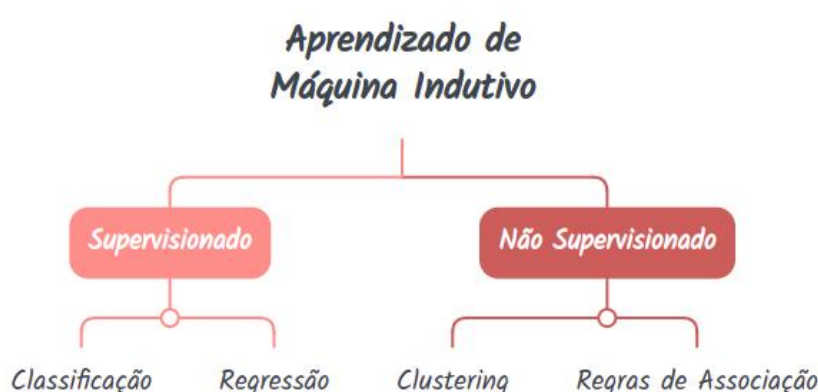


Figura 3 – Tipos de Aprendizado de Máquina.

Fonte: O Autor

Abaixo veremos a definição de alguns conceitos e termos bastante utilizados na literatura

de AM, com base em Sistemas Inteligentes ([MONARD; BARANAUSKAS, 2003](#)):

- **Indutor** é quem gera o classificador a partir da análise dos dados de entrada fornecidos, estes dados podem estar classificados (aprendizado supervisionado) ou não (aprendizado não-supervisionado). Após termos o algoritmo classificador, podemos utilizá-lo para a execução do conjunto de testes, obtendo assim a precisão do mesmo. Nós já sabemos qual é a classe de cada um dos elementos do conjunto de testes, sendo assim, quando fornecemos estes dados ao algoritmo classificador, podemos verificar a quantidade de acertos e erros.
- **Exemplo** consiste em uma lista de valores de atributos que irá descrever o objeto de interesse.
- **Atributo** carrega uma característica de um exemplo. O atributo pode ser classificado em dois tipos: o atributo contínuo, quando os valores podem ser ordenados (por exemplo, idade, um número natural) e nominal, quando os valores não podem ser ordenados (por exemplo, cor: preto, lilás, amarelo). É importante salientar que também temos a possibilidade de não possuímos um valor para um atributo do exemplo, ou até mesmo ele pode não se aplicar. Quando temos um atributo que só se aplica ao gênero masculino, nos exemplos do sexo feminino esse atributo não se aplicará. Para representar os casos em que o atributo é vazio, utilizamos uma interrogação “?”, e para representar os casos em que não se aplica utilizamos uma exclamação “!”. É extremamente necessário analisarmos quais atributos fazem sentido para o problema em questão, de forma a selecionarmos aqueles com alto poder preditivo, ou seja, que aumentarão a chance do sistema aprender corretamente.
- **Classe** é o principal interesse do desenvolvimento dos algoritmos de AM, ela faz parte do conjunto de atributos que cada exemplo possui. Sua importância se deve ao fato de que o objetivo do classificador é, a partir dos demais atributos do exemplo, conseguirmos encontrar a que classe ele pertence, se estivermos lidando com classificação; ou definir um valor real, no caso de regressão.
- **Conjunto de Exemplos** pode ser definido como sendo um conjunto de tuplas que contém uma lista de atributos e uma classe associada. Comumente, dividimos esse conjunto em duas partes muito importantes no contexto do desenvolvimento dos algoritmos, o conjunto de treinamento e o conjunto de teste, sendo que necessariamente os conjuntos devem ser diferentes, pois o conjunto de treinamento será utilizado para a geração do classificador, e o conjunto de testes para verificarmos a acurácia do classificador gerado.
- **Ruído** é o nome utilizado para definir a imperfeição dos dados que trabalhamos, algo que é extremamente comum de ser observado. Um exemplo clássico de ruído é quando possuímos dois elementos com os mesmos valores de atributos, porém com classes diferentes. A origem destes problemas nos dados podem ser as mais diversas possíveis.

- **Conhecimento do Domínio** diz respeito ao conhecimento sobre os valores válidos dos atributos escolhidos.
- **Classificador** é o resultado que o indutor irá gerar a partir do conjunto de treinamento, sendo que, ao informarmos para o classificador um novo exemplo, o mesmo conseguirá prever com uma excelente precisão qual é a classe do mesmo.
- **Viés** é o ato de induzirmos o algoritmo a preferir uma hipótese sobre outra. Os atributos em muitos casos irão possuir pesos diferentes, dessa forma estamos enviesando o mesmo a preferir alguns atributos sobre outros, algo que é de certa forma delicado, pois em algumas situações fazem com que o aprendizado tenha certas tendências (CARVALHO et al., 2021).
- **Modo de Aprendizado** pode ser dividido em incremental e não-incremental. O modo não-incremental é caracterizado quando precisamos que todo o conjunto de treinamento esteja presente no desenvolvimento do aprendizado; já no conjunto incremental não, precisamos apenas da hipótese antiga para atualizá-la assim que novos exemplos são acrescentados no conjunto de treinamento.

A Figura 4 mostra as etapas para aplicação das técnicas do aprendizado de máquina.

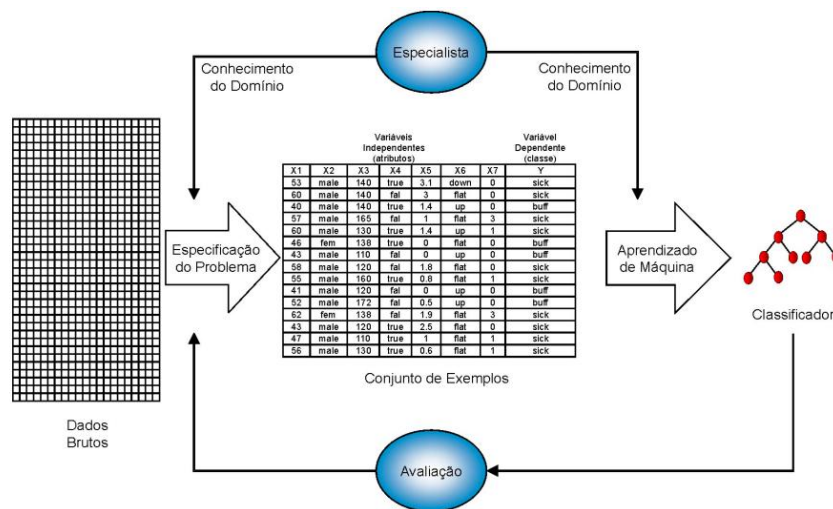


Figura 4 – Etapas para aplicação de técnicas de AM.

Fonte: (MONARD; BARANAUSKAS, 2003)

2.1.1 Paradigmas de Aprendizado de Máquina

Existem cinco principais paradigmas de aprendizado de máquina, segundo (MONARD; BARANAUSKAS, 2003), como veremos a seguir: baseado em exemplos, conexionista, estatístico, genético e simbólico.

2.1.1.1 Baseado em Exemplos

Este paradigma consiste em classificarmos um exemplo através de outros similares. Assumimos que dois exemplos com características iguais ou muito similares terão a mesma classificação. Denominamos esse tipo de aprendizado preguiçoso (do inglês, *lazy*).

Em aprendizados deste tipo, precisamos memorizar os exemplos já classificados para classificarmos os próximos. Um dos algoritmos mais conhecidos neste paradigma é o *K-Nearest Neighbours* (KNN ou K-Vizinhos mais próximos).

2.1.1.2 Conexionista

O nome deste modelo de aprendizado está diretamente relacionado ao fato dele envolver diversas unidades altamente conectadas. O exemplo mais conhecido de aprendizado de máquina conexonista são as redes neurais, que serão abordadas mais adiante.

2.1.1.3 Estatístico

Utiliza modelos estatísticos para a elaboração de classificadores. Para alguns autores, as redes neurais também se encaixam no paradigma de aprendizado estatístico, já que precisamos encontrar os valores apropriados para os pesos das conexões da rede e o viés.

2.1.1.4 Genético

Paradigma baseado na teoria de Darwin que diz que os seres que se adaptam melhor ao ambiente sobrevivem. Esta relação pode ser observada pelo fato do classificador genético consistir em uma população de classificadores que competem entre si para fazer previsões, sendo que os que obtêm os melhores resultados se multiplicam, e aqueles com resultados ruins são descartados.

2.1.1.5 Simbólico

A partir da análise de exemplos e contra-exemplos de um determinado conceito, constituem representações através de símbolos para os mesmos. Em linhas gerais essas representações estão na forma de expressões lógicas, redes semânticas e árvores de decisão.

2.2 Redes Neurais

Uma rede neural é um conjunto de elementos de processamento simples, denominados nós, que possuem funcionalidades baseadas nas de um neurônio animal. A capacidade de processamento da rede como um todo é depositada nas forças de conexão entre essas unidades, na literatura é comum essa força também ser denominada *peso* entre as conexões, que são obtidos por

um processo de adaptação ou aprendizado de um conjunto de dados de treinamento (GURNEY, 1997).

2.2.1 Neurônio Biológico

O aprofundamento do conhecimento sobre a constituição e o comportamento dos seres vivos tem possibilitado a criação de algoritmos bioinspirados, que se baseiam nos padrões e nas características observadas na natureza. As redes neurais tiveram um processo similar em seu desenvolvimento, pois espelhamos, em partes, o comportamento de um cérebro humano, mais especificamente os neurônios que o compõem. Sendo assim, é necessário entendermos alguns conceitos básicos sobre o funcionamento do cérebro humano antes de ingressarmos na computação de fato.

O cérebro humano é uma das estruturas mais complexas existente conhecida, ele possui cerca de 100 bilhões de neurônios. Eles se comunicam através de sinais elétricos, impulsos de curta duração, e as conexões entre neurônios, também denominadas interneurônios, são mediadas por junções eletroquímicas chamadas sinapses, que estão localizadas em ramos da célula chamados dendritos. Cada neurônio, que possui uma estrutura similar à representação da Figura 5, recebe milhares de conexões de outros neurônios, e, portanto, está constantemente recebendo uma infinidade de sinais de entrada, que eventualmente atingem o corpo celular. Quando isto acontece, os neurônios são integrados ou somados de alguma forma, e se o sinal resultante exceder seu limite, o neurônio irá disparar um impulso nervoso como resposta. Esta resposta é transmitida para outros neurônios através de uma fibra ramificada conhecida como axônio (GURNEY, 1997).

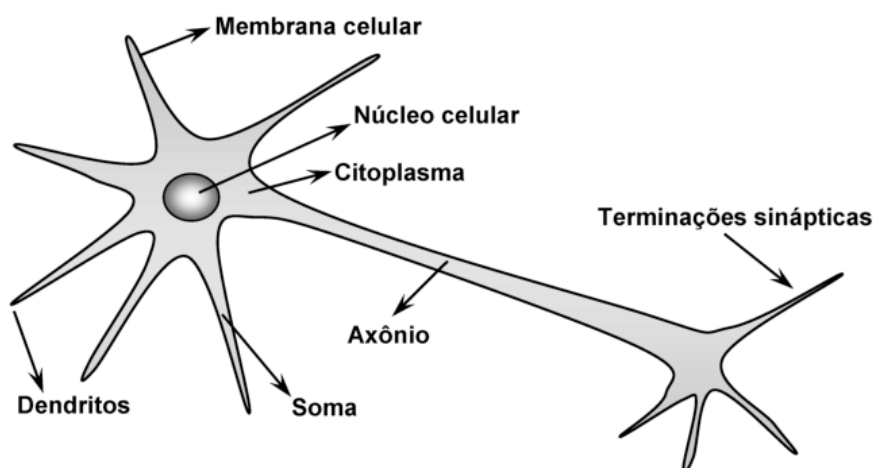


Figura 5 – Neurônio Biológico.

Fonte: (TEODORO; GOMES, 2018)

Para determinar se um impulso deve ser produzido ou não, alguns sinais de entrada produzem um efeito inibitório e tendem a impedir o disparo do mesmo, enquanto outros são excitatórios e promovem a geração desses impulsos. A capacidade de processamento de cada um dos neurô-

nios, portanto, está atrelada ao seu tipo, que será excitatório ou inibitório, além da força de suas conexões sinápticas com outros neurônios. Tratando-se agora dos neurônios artificiais, que veremos a seguir, o objetivo é implementarmos essa arquitetura conhecida de um neurônio biológico e estilo de processamento para produzirmos os mesmos efeitos computacionalmente ([GURNEY, 1997](#)).

2.2.2 Neurônio Artificial

São três os elementos básicos que precisamos espelhar para simular o comportamento de um neurônio biológico, que possuem uma estrutura similar a representada na Figura 6: o conjunto de conexões, um somador e uma função de ativação. Abaixo temos a definição de cada um segundo [Haykin \(2011\)](#).

- As **conexões** entre os nós possuirão um peso, como visto anteriormente. Ou seja, um sinal x_j na entrada da sinapse j conectado a um neurônio k deve ser multiplicada pelo peso entre as conexões (peso sináptico) w_{kj} . Um ponto que vale ser ressaltado é que diferentemente do que ocorre com os neurônios biológicos, o neurônio artificial poderá ter peso sináptico negativo.
- O **somador** irá somar todos os sinais de entrada ponderados com seus respectivos pesos sinápticos, resultando em uma combinação linear.
- Ao final, a **função de ativação** atuará limitando a amplitude do sinal de saída para que ele seja um valor finito, por este motivo ela também pode ser chamada na literatura de função de esmagamento.

O termo rede será usado neste trabalho para se referir a qualquer sistema de neurônios artificiais. Isso pode variar de algo tão simples como um único nó até uma grande coleção de nós em que cada um está conectado a todos os outros nós da rede.

Nos neurônios reais, os pesos sinápticos podem, de acordo com determinadas circunstâncias, ser modificados de modo que o comportamento de cada neurônio possa mudar ou se adaptar à sua entrada de estímulo particular. Em neurônios artificiais, o equivalente a isso é a modificação dos valores de peso. Todo o conhecimento que a rede possui deve ser armazenado em seus pesos, que evoluem por um processo de adaptação ao estímulo de um conjunto de exemplos de padrões. As redes neurais fazem uso do aprendizado de máquina supervisionado, e a cada atualização dos pesos sinápticos, é como se a rede estivesse adquirindo um novo aprendizado.

Após atualizações de peso, outro exemplo é apresentado, a saída comparada com a classe definida pelo especialista e novas alterações feitas. Essa sequência de eventos é repetida iterativamente muitas vezes até que o comportamento da rede convirja para que sua resposta a cada

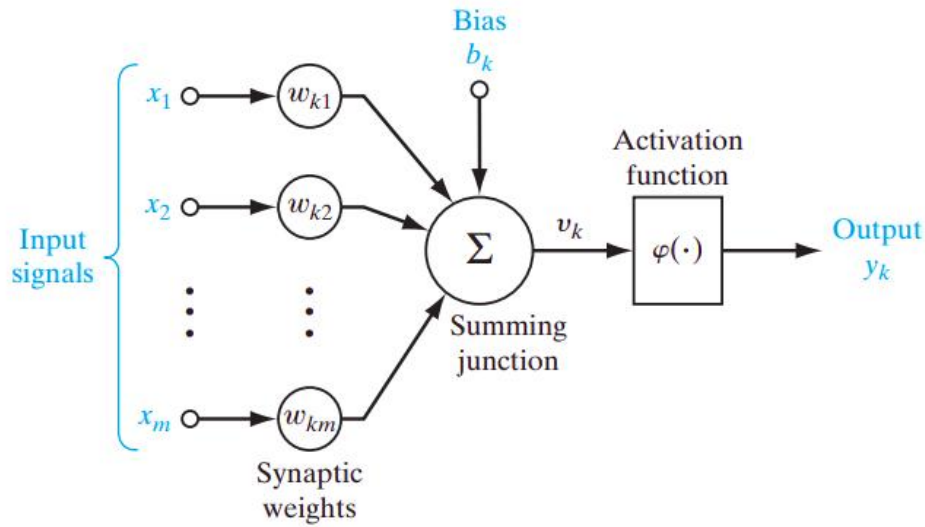


Figura 6 – Neurônio Artificial.

Fonte: (HAYKIN, 2011)

exemplo esteja próxima do alvo correspondente. O processo como um todo, incluindo qualquer ordenação de apresentação de padrões e critérios para encerrar o processo, constitui o algoritmo de treinamento. Após o treinamento, se apresentarmos à rede um exemplo que ela não viu anteriormente, caso a mesma tenha aprendido corretamente a estrutura do domínio do problema, então ela deve classificar o exemplo desconhecido corretamente (GURNEY, 1997).

O modelo de neurônio artificial da Figura 6 também contém um viés que é aplicado externamente, denotado por b_k . Ele tem o objetivo de oscilar a entrada de nossa função de ativação, aumentando ou diminuindo a mesma, dependendo de ser positiva ou negativa, respectivamente. Podemos descrever o neurônio representado na figura através das seguintes equações (HAYKIN, 2011):

$$u_k = \sum_{j=1}^m w_{kj} x_j, \quad (2.1)$$

e

$$y_k = \varphi(u_k + b_k), \quad (2.2)$$

onde k representa um neurônio, os sinais de entrada são representados por x_1, x_2, \dots, x_m ; os pesos das sinapses são dados por $w_{k1}, w_{k2}, \dots, w_{km}$; u_k representa o resultado da combinação linear de acordo com os sinais de entrada fornecidos; o viés é representado por b_k ; a multiplicação representa a função de ativação; e por fim, y_k é o sinal de saída do neurônio. Para conseguirmos aplicar a transformação afim à saída u_k do combinador linear, se faz necessária a utilização do viés, conforme mostrado por 2.3.

$$v_k = u_k + b_k. \quad (2.3)$$

2.2.3 Funções de Ativação

A função de ativação é quem definirá a saída do neurônio em relação ao campo local induzido v . Existem dois tipos básicos de funções de ativação que veremos em sequência: Função com Limite e Função Sigmoidal.

2.2.3.1 Função com Limite (*Threshold*)

A função com limite também é comumente chamada de função *Heaviside*. Ela é expressa da seguinte forma:

$$\varphi(v) = \begin{cases} 1, & v \geq 0 \\ 0, & v < 0 \end{cases} \quad (2.4)$$

Os neurônios que utilizam essa função de ativação são chamados de modelo McCulloch-Pitts, uma homenagem ao estudo realizado por McCulloch e Pitts em 1943. No modelo em questão, a saída de um determinado neurônio assume o valor 1 caso seu campo local induzido for maior ou igual a zero, e assume valor 0 caso contrário. Esta declaração explicita uma propriedade chamada *tudo-ou-nada*, proposta no modelo McCulloch-Pitts (HAYKIN, 2011). Podemos visualizar um exemplo de gráfico gerado por essa função na Figura 7.

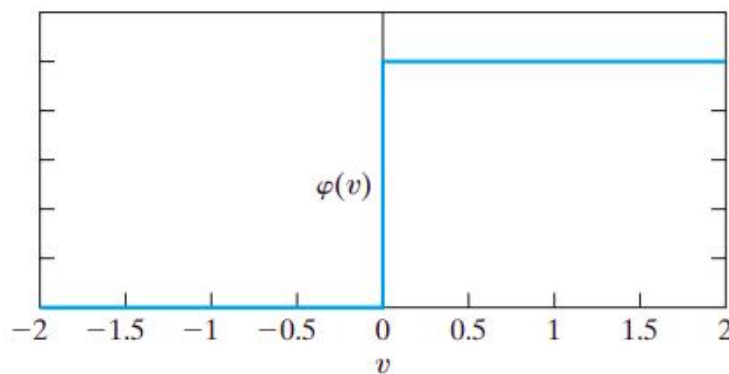


Figura 7 – Exemplo de gráfico de Função com Limite.

Fonte: (HAYKIN, 2011)

2.2.3.2 Função Sigmoidal

A função sigmoide, que possui um gráfico em forma de *S*, como mostrado na Figura 8, é um tipo de função de ativação muito popular no desenvolvimento de redes neurais. Uma de suas importantes propriedades é o fato de ser definida como uma função estritamente crescente,

ou seja, uma função que exibe um equilíbrio entre o comportamento linear e não linear (HAYKIN, 2011). A função logística é um exemplo de função sigmoide, definida por:

$$\varphi(v) = \frac{1}{1 + \exp(-av)}, \quad (2.5)$$

onde a é quem determina sua inclinação. Podemos obter funções sigmoides de diferentes inclinações variando esse parâmetro. No limite, à medida que a tende ao infinito, a função de ativação do tipo sigmoide acaba se tornando simplesmente uma função de limiar. Tratando-se de uma função limiar, ela assumirá o valor de 0 ou 1, diferentemente do efeito causado pela função sigmoide, que pode assumir uma faixa contínua de valores entre 0 e 1. Outro fator importante que vale salientarmos é que a função sigmoide é diferenciável, enquanto a função limiar não (HAYKIN, 2011).

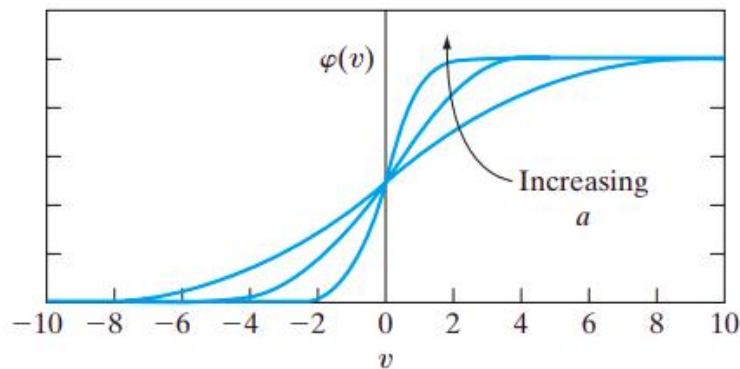


Figura 8 – Exemplo de gráfico de Função Sigmoide.

Fonte: (HAYKIN, 2011)

2.3 Aprendizado Profundo

Antes de começarmos a falar sobre aprendizado profundo efetivamente, é interessante conhecermos qual foi a motivação para o seu surgimento. Os primeiros sucessos alcançados na área de IA possuíam um atributo em comum, todos eles foram alcançados em ambientes determinísticos, aqueles em que não temos o fator da imprevisibilidade e a saída obtida depende única e exclusivamente da ação de um agente (GOODFELLOW; BENGIO; COURVILLE, 2016a).

Um exemplo muito famoso de um dos sucessos obtidos nessa área aconteceu no ano de 1997, quando de maneira surpreendente para a época, o sistema de jogo de xadrez *Deep Blue*, desenvolvido pela *International Business Machines Corporation* (IBM), derrotou o então campeão mundial de xadrez, o grande mestre Garry Kasparov (CAMPBELL; JR; HSU, 2002). O xadrez é considerado um problema simples de ser resolvido pelo computador por se tratar de um sistema fechado, já que conseguimos calcular todas as jogadas possíveis de serem feitas - o número total de jogadas possíveis no xadrez é maior que o número estimado de átomos no universo.

A grande dificuldade dos computadores, de forma até mesmo irônica, está nas tarefas mais simples para os seres humanos, como reconhecer rostos, objetos ou fala, pois são tarefas que exigem um conhecimento amplo sobre o mundo. Isso pode se tornar extremamente dificultoso para as máquinas pelo fato de grande parte desse conhecimento ser subjetivo e intuitivo, portanto, difícil de formalizarmos. A inteligência artificial surgiu com o intuito de facilitar a transferência do conhecimento informal nos computadores. Embora as máquinas tenham superado os humanos em jogos de xadrez há algum tempo, somente recentemente começaram a avançar no que diz respeito ao conhecimento informal (GOODFELLOW; BENGIO; COURVILLE, 2016a).

O Aprendizado Profundo, ou *Deep Learning*, é um subconjunto do Aprendizado de Máquina, com modelos baseados em redes neurais de muitas camadas - o que justifica o *deep* no nome. A principal diferença do aprendizado profundo é que as operações que são aprendidas em cada uma das diversas camadas de representações da rede, são aprendidas em conjunto com os dados. Sendo assim, problemas em que os métodos tradicionais de rede neural falham, como processamento de imagens ou sinal de áudio, as redes neurais profundas se destacam.

Apesar de ser um modelo que constrói conceitos complexos, isso é feito a partir de conceitos mais simples. A Figura 9 mostra um exemplo de um sistema de aprendizado profundo. No exemplo em questão, a rede recebe uma imagem, representada por uma série de pixels como entrada, que é apresentada à camada visível. Logo em seguida, as camadas ocultas, que carregam esse nome pelo fato de seus valores não serem fornecidos, irão extrair recursos ainda mais abstratos da imagem de entrada. Por fim, a descrição da imagem gerada nas camadas ocultas em termos das partes dos objetos que ela possui, pode ser usada no reconhecimento dos mesmos, como podemos observar na saída (GOODFELLOW; BENGIO; COURVILLE, 2016a).

Um exemplo de algoritmo que aplica esses conceitos vistos é o *Multilayer Perceptron* (MLP), que se trata de uma função matemática formada pela composição de inúmeras funções mais simples, onde a saída de uma camada que executará uma dessas funções servirá de entrada para a próxima. A profundidade da rede determina a quantidade de instruções em sequência que serão executadas.

2.3.1 Redes Neurais Convolucionais

Uma Rede Neural Convolucional, ou *Convolutional Neural Network* (CNN), é um tipo de rede de aprendizado profundo. Essa arquitetura de rede neural é utilizada na aplicação de matrizes bidimensionais, como imagens, por exemplo. A CNN surgiu na década de 1960, com ideias como percepção local, pesos do compartilhamento e amostragem no espaço ou no tempo. O benefício das CNNs é que são mais fáceis de treinar e têm muito menos parâmetros do que redes totalmente conectadas com a mesma quantidade de unidades ocultas (DUMOULIN; VISIN, 2016).

A arquitetura de rede neural de convolução é geralmente usada em colaboração com a camada de convolução e a camada de *pool*, que serão explicadas adiante (LECUN; KAVUK-

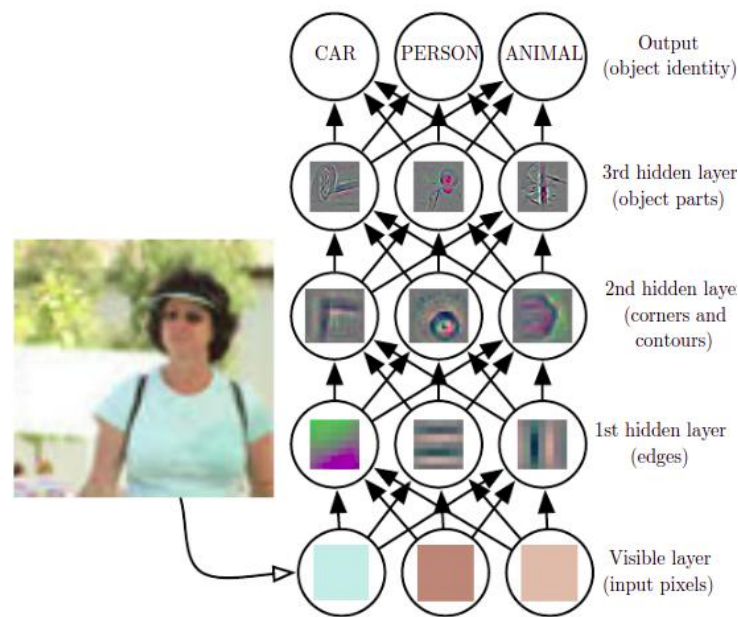


Figura 9 – Exemplo de um modelo de aprendizado profundo.

Fonte: (GOODFELLOW; BENGIO; COURVILLE, 2016a)

CUOGLU; FARABET, 2010). A arquitetura da CNN é composta por vários mapas; cada mapa é composto de várias unidades neurais; todas as unidades neurais no mesmo mapa compartilham um *kernel* de convolução e cada *kernel* de convolução representa um recurso, como acesso à borda da imagem. O recurso de imagem de convolução multiescala é gerado definindo o tamanho e o parâmetro do *kernel* de convolução. A informação de diferentes ângulos é gerada no espaço de características (DUMOULIN; VISIN, 2016). Na Figura 10 podemos observar o exemplo de uma CNN aplicada ao processamento de imagens de tomografia torácica para a classificação de um tumor como maligno ou benigno.

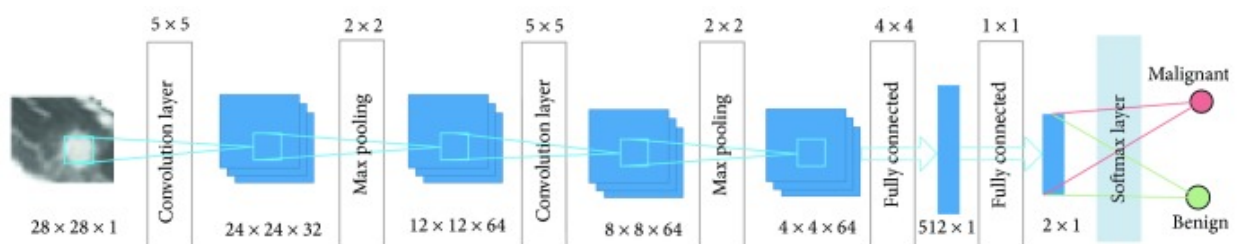


Figura 10 – Exemplo de um modelo de CNN.

Fonte: (SONG et al., 2017)

2.3.1.1 Convolução

Um recurso extremamente importante utilizado pelas CNNs são as transformações afins: um vetor é recebido como entrada e é multiplicado por uma matriz para produzir uma saída. Isso

é aplicável a qualquer tipo de entrada, como uma imagem, por exemplo, e para qualquer que seja sua dimensionalidade, sua representação sempre pode ser achatada em um vetor antes das transformações. Imagens, cliques de som e muitos outros tipos semelhantes de dados têm uma estrutura intrínseca. Mais formalmente, eles compartilham estas propriedades importantes (DUMOULIN; VISIN, 2016):

- Eles são armazenados como arrays multidimensionais;
- Eles apresentam um ou mais eixos para os quais a ordenação é importante.

Essas propriedades não são exploradas quando uma transformação afim é aplicada; de fato, todos os eixos são tratados da mesma forma e a informação topológica não é levada em consideração. Ainda assim, aproveitar a estrutura implícita dos dados pode ser muito útil na resolução de algumas tarefas, como visão computacional e reconhecimento de fala, e nesses casos seria melhor preservá-la. Uma convolução discreta é uma transformação linear com a capacidade de preservar essa noção de ordenação. Na Figura 11, é exibido um exemplo do processo de convolução de uma rede (DUMOULIN; VISIN, 2016).

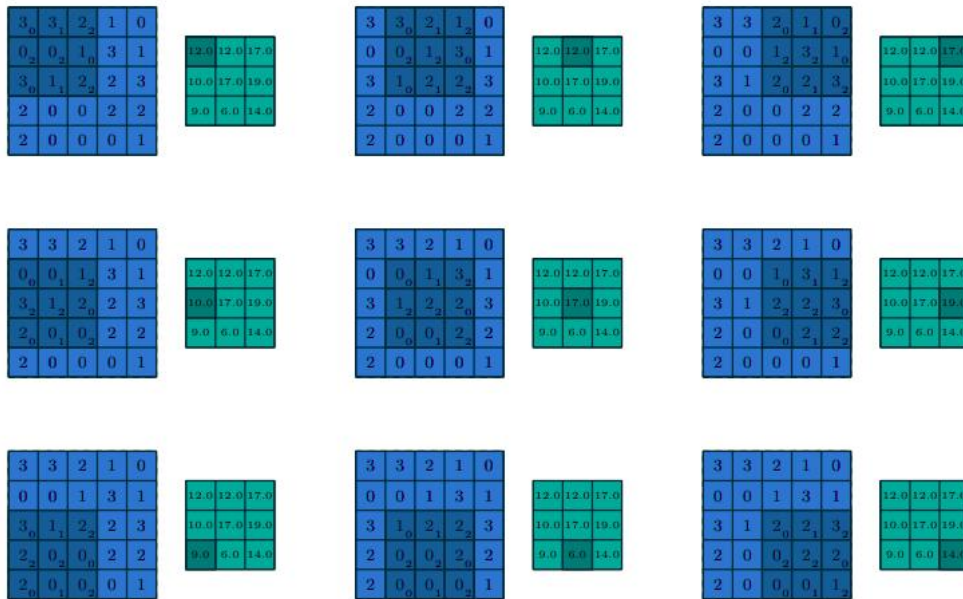


Figura 11 – Demonstração da camada de Convolução.

Fonte: (DUMOULIN; VISIN, 2016)

A coleção de *kernels* que definem uma convolução discreta tem uma forma correspondente a alguma permutação de (n, m, k_1, \dots, k_N) , onde

- n = número de mapas de recursos de saída;
- m = número de mapas de recursos de entrada;

- k_j = tamanho do *kernel* ao longo do eixo j .

As seguintes propriedades afetam o tamanho de saída o_j de uma camada convolucional ao longo do eixo j :

- i_j = tamanho de entrada ao longo do eixo j ;
- k_j = tamanho do *kernel* ao longo do eixo j ;
- s_j = passo (distância entre duas posições consecutivas do *kernel*) ao longo do eixo j ;
- p_j = preenchimento de zeros (número de zeros concatenados no início e no final de um eixo) ao longo do eixo j .

2.3.1.2 Pooling

Outra importante operação executada pelas CNNs são as camadas de *pooling*. Elas são responsáveis pela redução do tamanho dos mapas de recursos através do uso de alguma função para resumir as sub-regiões, como obter a média ou o valor máximo.

O *pooling* funciona deslizando uma janela pela entrada e alimentando o conteúdo da janela para uma função de *pooling*. De certa forma, o *pooling* funciona como uma convolução discreta, mas substitui a combinação linear descrita pelo *kernel* por alguma outra função. A Figura 12 exibe um exemplo de *pooling*. O tamanho de saída o_j de uma camada de agrupamento ao longo do eixo j é afetada pelas seguintes propriedades:

- i_j : tamanho de entrada ao longo do eixo j ;
- k_j : tamanho da janela de agrupamento ao longo do eixo j ;
- s_j : passo ao longo do eixo j .

2.3.1.3 VGG

As *Very Deep Convolutional Networks* (VGG), ou Redes Neurais Muito Profundas, isto é, com muitas camadas, foram desenvolvidas com o objetivo de aumentar a profundidade das CNNs, a fim de aumentar também o desempenho desse modelo.

A arquitetura dessa rede serve de base para o reconhecimento de objetos inovadores, sendo uma das arquiteturas mais populares no reconhecimento de imagens. Os dois tipos mais comuns de redes VGG são as redes VGG16 e VGG19, o número atrelado a cada uma delas está relacionado a quantidade máxima de camadas que a rede suporta. No artigo de [Simonyan e Zisserman \(2014\)](#), a VGG foi testada com a utilização do conjunto de dados *ImageNet*, composto por mais de 14

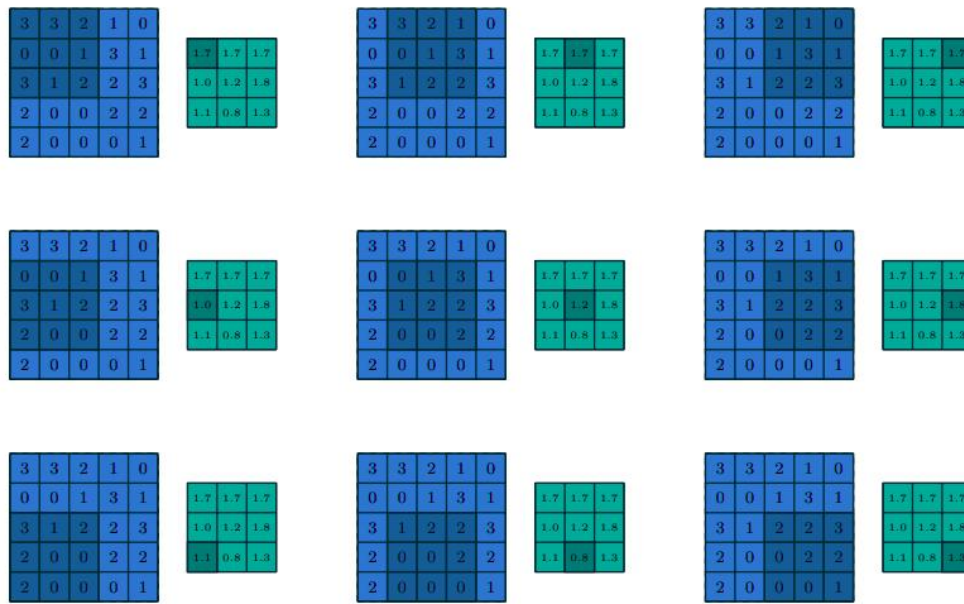


Figura 12 – Demonstração da camada de Pooling.

Fonte: (DUMOULIN; VISIN, 2016)

milhões de imagens divididas em cerca de 1.000 classes. O modelo VGG16 atingiu a marca de aproximadamente 92,7% de acurácia.

Na Figura 13, conseguimos ver um comparativo da arquitetura das redes VGG, de acordo com sua quantidade de camadas, que nos testes realizados por Simonyan e Zisserman (2014), variaram a rede de 11 a 19 camadas.

2.3.1.4 Resnet

A *Residual Neural Network* (Resnet), ou Rede Neural Residual, proposta por He et al. (2016), faz o uso de conexões de salto para reduzir, de forma significativa, a dificuldade de treinamento, o que resultou em grandes aumentos de desempenho em termos de erro de treinamento e generalização.

Os autores seguiram uma abordagem baseada em princípios para adicionar conexões de salto a cada duas camadas a uma rede no estilo VGG (SIMONYAN; ZISSERMAN, 2014). Após uma observação empírica de He et al. (2016), foi analisado que os saltos deveriam ter comprimento dois, já que os testes com outros comprimentos tiveram uma eficiência menor, pois saltos de maior profundidade resultam em um ponto estacionário extremamente plano, do qual o algoritmo de otimização é difícil de escapar. O salto 1, no entanto, é essencialmente equivalente a nenhum salto. Como o número de camadas tende ao infinito, basta observar apenas a função de perda no ponto inicial zero.

A Resnet tornou o treinamento mais fácil, conseguindo ainda atingir uma quantidade me-

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figura 13 – Comparativo de redes do tipo VGG.

Fonte: (SIMONYAN; ZISSERMAN, 2014)

nor de erros de treinamento e teste. Fazendo uso dessa nova estrutura, He et al. (2016) conseguiram treinar uma rede com 1001 camadas, feito esse que anteriormente era considerado praticamente impossível. Foi descoberto que, mantendo um caminho de salto limpo, as redes residuais tem um desempenho ainda melhor. He et al. (2016) sugerem que as camadas em redes residuais estão aprendendo mapeamentos residuais, tornando-os mais fáceis de representar mapeamentos de identidade, o que evita que as redes se degradem quando as profundidades das redes aumentam. Na Figura 14 podemos observar um comparativo entre as arquiteturas das redes VGG19 com 34 camadas e Resnet com 34 camadas.

2.3.1.5 Xception

A rede neural convolucional Xception, ou *Extreme Inception*, baseia-se em camadas de convolução que são separáveis em profundidade. Essa arquitetura é baseada na seguinte hipótese: que o mapeamento de correlações de canais cruzados e correlações espaciais nos mapas de características de redes neurais convolucionais podem ser totalmente desacoplados. Como essa hipótese é uma versão mais robusta da hipótese subjacente à arquitetura Inception, a nova arquitetura proposta foi denominada Xception (CHOLLET, 2017).

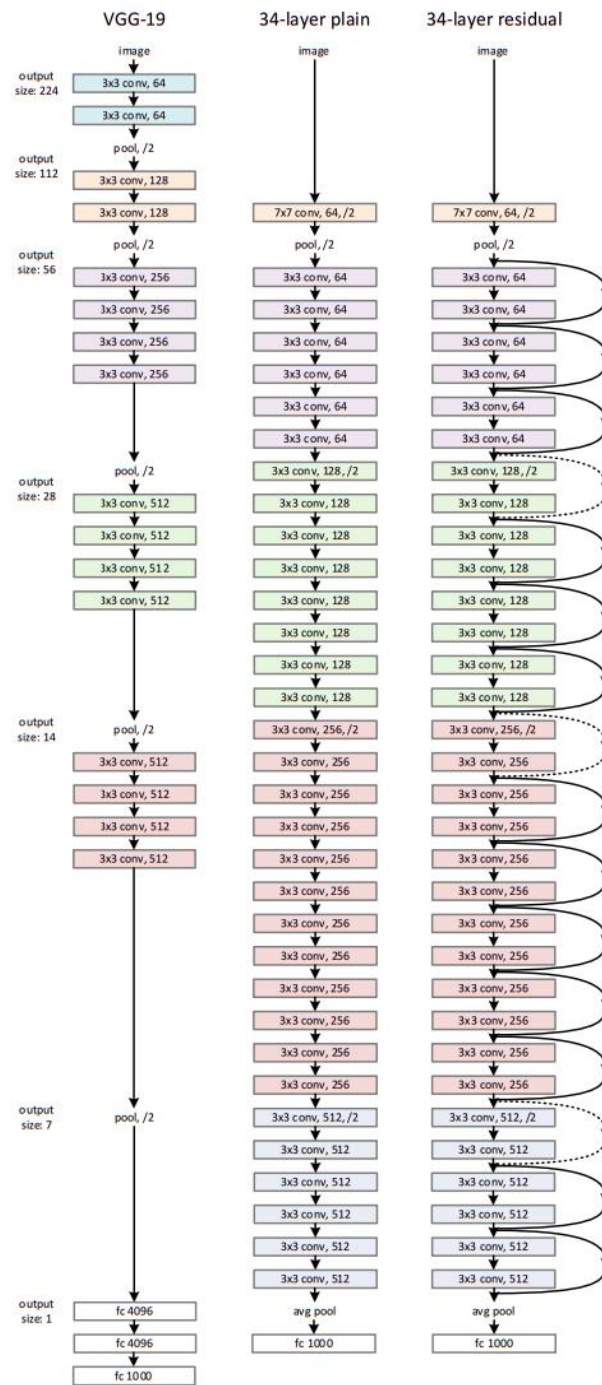


Figura 14 – Arquitetura das redes VGG19 e Resnet.

Fonte: (HE et al., 2016)

A arquitetura Xception, exibida na Figura 15, possui 36 camadas de convolução que formam um centro de extração de características da rede. De forma opcional, pode-se inserir camadas totalmente conectadas antes da camada de regressão logística. As 36 camadas convolucionais estão estruturadas em 14 módulos, e cada um desses módulos, com exceção do primeiro e do último, possuem conexões residuais lineares no seu entorno. Pelo fato desse modelo se basear em cama-

das convolucionais separáveis, isso torna a arquitetura muito fácil de ser definida ou modificada, utilizando poucas linhas de código (CHOLLET, 2017).

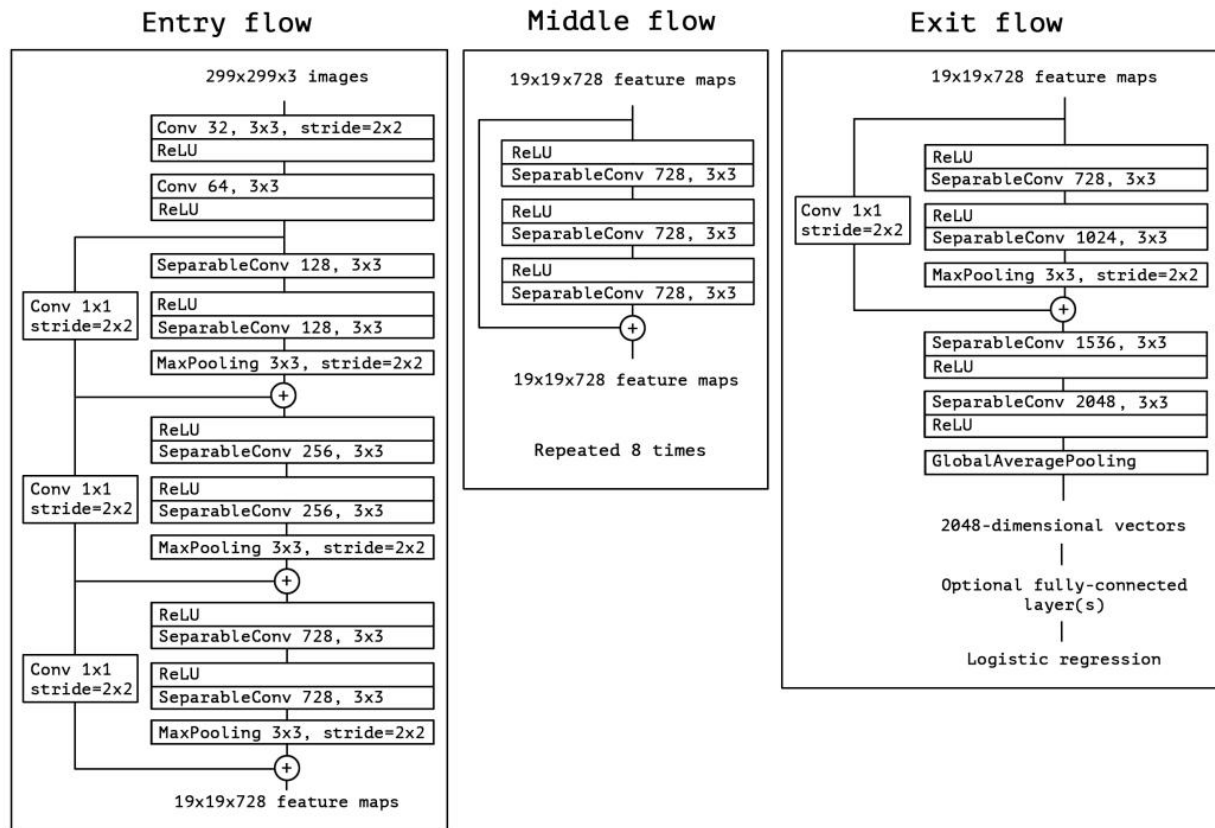


Figura 15 – Arquitetura da rede Xception.

Fonte: (CHOLLET, 2017)

2.4 Processamento de Imagens

O processamento de imagens possui como objetivo o tratamento dos dados que uma imagem carrega para uma análise, seja ela computacional ou humana (GONZALEZ; WOODS, 2008). Nos dias atuais existem diversas aplicações para este processamento, como interpretação de imagens médicas, como o raio-X, por exemplo; reconhecimento de padrões, no aprendizado de máquina, além da contribuição na área ambiental com o monitoramento de queimadas. O aumento da quantidade de aplicações é diretamente proporcional ao aumento da capacidade dos computadores que possuímos e da diminuição de seus preços, permitindo com que mais pessoas tenham acesso a estes recursos.

São três as categorias fundamentais na área do processamento de imagens: realce, restauração e análise.

- **Realce** se trata de técnicas para melhorar a qualidade de uma imagem, possibilitando a sua utilização para uma ocasião específica;
- **Restauração** diz respeito à tentativa de recuperação de uma imagem que por algum motivo foi danificada;
- **Análise** está ligada ao reconhecimento de padrões para a extração de características ou informações das imagens.

Para determinadas aplicações, se faz necessário a utilização de mais de uma categoria fundamental para uma solução mais apropriada. É válido ressaltarmos que para podermos processar imagens, primordialmente necessitamos digitalizá-las. A seguir veremos alguns conceitos básicos sobre imagens digitais que serão necessários conhecermos para o pleno entendimento deste trabalho.

2.4.1 Representação Digital de Imagens

Uma imagem monocromática, ou seja, que apresenta apenas uma cor, é descrita como uma função bidimensional $f(x, y)$ onde x e y são coordenadas espaciais, e f se refere ao brilho da imagem no ponto em questão. Este ponto (x, y) é chamado de pixel. A Figura 16 exibe os eixos x e y posicionados na imagem.



Figura 16 – Representação de Imagem Digital.

Fonte: (GONZALEZ; WOODS, 2008)

A função f citada acima é o produto de duas outras funções, a função $i(x, y)$ que calcula a iluminação de uma imagem e a função $r(x, y)$, que calcula a reflectância da imagem, ou seja, a quantidade de luz que é refletida no ponto (x, y) (GONZALEZ; WOODS, 2008). Nas imagens coloridas, a representação é muito similar, mudando apenas o fato de cada um dos pixels ser representado como um vetor utilizando-se do sistema de cores RGB (*Red, Green and Blue*):

$$f(x, y) = (fR(x, y), fG(x, y), fB(x, y)), \quad (2.6)$$

onde as componentes fR , fG e fB representam as intensidades, para o ponto (x, y) , das cores vermelho, verde e azul. O que nos leva a afirmar que uma imagem colorida pode ser definida como sendo a união de três imagens monocromáticas (QUEIROZ; GOMES, 2006).

2.4.2 Pixels

O pixel é o elemento mais básico que compõe uma imagem e pode ser representado por uma coordenada (x, y) . O processamento de uma imagem é realizado a partir da manipulação de seus pixels, por este motivo se faz necessário conhecermos alguns de seus conceitos.

Começando pelo conceito de vizinhança, que nos diz que um pixel p possui o que chamamos de vizinhança-4, expressa na Figura 17, que faz referência aos pixels que circundam p na horizontal e vertical, e a vizinhança-8, expressa na Figura 18, que engloba a vizinhança-4, acrescentados dos 4 pixels adjacentes a p nas diagonais (GONZALEZ; WOODS, 2008).

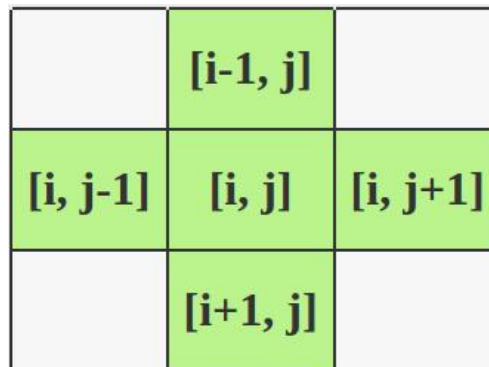


Figura 17 – Representação da Vizinhança-4 de um Pixel.

Fonte: O Autor

Também temos o conceito de conectividade, que diz que dois pixels distintos podem ser ditos conectados quando satisfizerem duas condições: são adjacentes e atendem a um critério específico de classificação (GONZALEZ; WOODS, 2008).

O caminho de um pixel p_1 para um pixel p_2 é a sequência de pixels que precisamos percorrer para chegar da origem (p_1) até o ponto de destino (p_2). Este conceito é importante para definirmos um componente conexo, que dado um subconjunto de pixels C de uma imagem e os pontos p_1 e p_2 que estão neste subconjunto C , podemos dizer que p_1 está conectado a p_2 apenas

$[i-1, j-1]$	$[i-1, j]$	$[i-1, j+1]$
$[i, j-1]$	$[i, j]$	$[i, j+1]$
$[i+1, j-1]$	$[i+1, j]$	$[i+1, j+1]$

Figura 18 – Representação da Vizinhança-8 de um Pixel.

Fonte: O Autor

se existir um caminho entre os dois pontos que só passa por elementos de S . Sendo assim, o componente conexo é dado pelo conjunto de pixels de C que estão conectados a p_1 (GONZALEZ; WOODS, 2008).

2.4.3 Etapas do Processamento de Imagens

Toda a jornada para se processar uma imagem é dividida em cinco etapas fundamentais: aquisição, pré-processamento, segmentação, representação e descrição, e por fim, reconhecimento e interpretação. Abaixo veremos uma breve explicação sobre cada uma delas:

- **Aquisição:** é onde capturamos a imagem por um sensor, posteriormente essa imagem é digitalizada através de um conversor analógico-digital. Feito isso, podemos manipular a imagem de maneira computadorizada;
- **Pré-processamento:** esta segunda etapa serve para melhoria das características e definição da imagem, além de atuar na redução dos ruídos (QUEIROZ; GOMES, 2006);
- **Segmentação:** nesta fase dividiremos a imagem em partes, processo este que irá variar de acordo com a finalidade da aplicação (GONZALEZ; WOODS, 2008). Será gerada então como saída um conjunto de pixels, que por sua vez representam o interior e fronteira de cada uma das partes divididas anteriormente;
- **Representação e descrição:** a forma de se exibir este conjunto de pixels gerados na fase anterior é chamada de representação, e após esta fase, é realizada a extração de informações úteis para facilitar a diferenciação das classes, fase denominada de descrição;
- **Reconhecimento e interpretação:** na etapa de reconhecimento, é atribuída uma classe para cada parte da imagem; e na etapa de interpretação, será associado um significado a cada uma das partes reconhecidas. Concluída todas as etapas, o resultado pode ser visualizado através de dispositivos de impressão.

2.5 Avaliação de Algoritmos

Após executarmos os algoritmos de redes neurais convolucionais no processamento das imagens de exames de tomografia computadorizada torácica, queremos comparar os resultados para descobrir qual algoritmo obteve os melhores resultados. Para tal, veremos nessa seção técnicas e medidas de avaliação de algoritmos de aprendizado de máquina.

2.5.1 Taxas de Acurácia e Erro

As taxas de acurácia e erro de um algoritmo estão relacionadas ao cálculo das porcentagens de falhas e acertos do classificador obtido no aprendizado de máquina, levando em conta problemas de classificação que será o tema abordado neste trabalho. A partir do conjunto de testes, descobrimos quais foram as porcentagens de classificações realizadas de maneira correta e incorreta, levando em consideração a quantidade total de dados no conjunto (MONARD; BARANAUSKAS, 2003).

A taxa de erro de um classificador k pode ser denotada como $err(k)$, que em linhas gerais pode ser obtida a partir da Equação 2.7, que irá comparar o rótulo que foi atribuído pelo classificador (k) com a classe verdadeira do exemplo (y) (MONARD; BARANAUSKAS, 2003).

$$err(k) = \frac{1}{n} \sum_{i=1}^n \|y_i \neq k(x_i)\|, \quad (2.7)$$

O operador $\|exp\|$ retorna 1 caso a expressão exp seja verdadeira, e 0 caso ela seja falsa. A variável n representa o número de exemplos dentro do contexto inserido. A precisão de nosso algoritmo, ou acurácia, pode ser obtida pela subtração demonstrada em 2.8 (MONARD; BARANAUSKAS, 2003).

$$acc(h) = 1 - err(k), \quad (2.8)$$

É trivial inferirmos que, quanto maior a acurácia de nosso classificador, melhor ele é.

2.5.2 Matriz de Confusão

A matriz de confusão nos fornece uma medida precisa do modelo de classificação, exibindo a quantidade de classificações corretas em comparação às classificações preditas para cada uma das classes de nosso domínio, um conjunto de exemplos E , por exemplo. A Tabela 1 mostra os resultados obtidos em 2 Dimensões (2D): classes verdadeiras e preditas, para n classes diferentes C_1, C_2, \dots, C_n (MONARD; BARANAUSKAS, 2003).

Cada um dos elementos $M(C_i, C_j)$ da matriz, para $i, j = 1, 2, \dots, n$ calculados através da Equação 2.9, onde o operador $\|exp\|$ retorna 1 caso a expressão exp seja verdadeira, e 0 caso ela

seja falsa, irão representar a quantidade de exemplos do conjunto E , citado anteriormente, que pertencem a classe C_i , mas que o classificador classificou como sendo da classe C_j (MONARD; BARANAUSKAS, 2003).

Tabela 1 – Exemplo de Matriz de Confusão

Classe	Predita C_1	Predita C_2	. . .	Predita C_n
Verdadeira C_2	$M(C_1, C_1)$	$M(C_1, C_2)$. . .	$M(C_1, C_n)$
Verdadeira C_1	$M(C_2, C_1)$	$M(C_2, C_2)$. . .	$M(C_2, C_n)$
.
.
.
Verdadeira C_K	$M(C_n, C_1)$	$M(C_n, C_2)$. . .	$M(C_n, C_n)$

$$M(C_i, C_j) = \sum_{\forall (x,y) \in E: y=C_i} \|k(x) = C_j\|, \quad (2.9)$$

Para simplificar o entendimento da matriz de confusão, podemos pensar no problema dividido em duas classes rotuladas como positivo (+) e negativo (−), que representam o acerto e erro do classificador, respectivamente. Na matriz mostrada na Tabela 1, a diagonal principal equivale ao número de acertos do algoritmo, mais precisamente a quantidade de elementos da classe positiva classificados corretamente o que denominaremos TP , derivado de *True Positive*, ou Verdadeiro Positivo, traduzindo para o português. A quantidade de acertos relativos a classe negativa será denominada TN , derivado de *True Negative*, ou Verdadeiro Negativo. A quantidade de erros de classificação das classes positivo e negativo, são denominados *False Positive (FP)* e *False Negative (FN)*, respectivamente. Sendo assim, o número total de exemplos pode ser obtido através do resultado da soma $n = (TP + FN + FP + TN)$.

Através desses valores conseguimos calcular outras medidas que colaboram na realização da análise de desempenho de um algoritmo, tais como a confiabilidade positiva - **prel** (*positive reliability*), a confiabilidade negativa - **nrel** (*negative reability*), o suporte - **sup**, a sensibilidade - **sens** (*sensitivity* ou também denominada *recall*), a especificidade - **spec** (*specificity*), cobertura - **cov** (*coverage*), e precisão total - **tacc** (*total accuracy*), que são calculadas através das fórmulas 2.10 à 2.16, respectivamente (WEISS; KULIKOWSKI, 1991).

$$prel(k) = \frac{TP}{TP + FP}, \quad (2.10)$$

$$nrel(k) = \frac{TN}{TN + FN}, \quad (2.11)$$

$$sup(k) = \frac{TP}{n}, \quad (2.12)$$

$$sens(k) = \frac{TP}{TP + FN}, \quad (2.13)$$

$$spec(k) = \frac{TN}{TN + FP}, \quad (2.14)$$

$$tacc(k) = \frac{TP + TN}{n}, \quad (2.15)$$

$$cov(k) = \frac{TP + FP}{n}. \quad (2.16)$$

2.5.3 Gradient-weighted Class Activation Mapping

Gradient-weighted Class Activation Mapping, ou Grad-CAM, é uma técnica de visualização que permite entender quais partes de uma imagem são mais relevantes para a classificação feita por uma rede neural convolucional. Ao contrário de outras técnicas de visualização, o Grad-CAM não requer acesso direto às camadas convolucionais, tornando-a uma abordagem mais genérica e aplicável a diferentes arquiteturas de redes neurais convolucionais.

O Grad-CAM usa os gradientes de ativação da camada de saída da rede neural em relação à classe de interesse para mapear as regiões ativadas na imagem de entrada. Esses gradientes são ponderados pela importância relativa de cada canal de ativação e, em seguida, somados para produzir o mapa de ativação da classe. Esse mapa é então sobreposto à imagem original para destacar as regiões que contribuíram significativamente para a classificação.

Essa técnica de visualização é útil para interpretar o processo de decisão das redes neurais convolucionais, fornecendo *insights* sobre quais características da imagem são consideradas relevantes para a classificação. O Grad-CAM tem aplicações em várias áreas, como medicina, visão computacional e análise de imagens, ajudando a compreender e explicar o funcionamento das redes neurais convolucionais de forma mais intuitiva (SELVARAJU et al., 2017).

3 Revisão Bibliográfica

Neste capítulo veremos alguns resultados de trabalhos já realizados na literatura explorando o tema da utilização de redes neurais aplicadas ao diagnóstico do câncer de pulmão.

3.1 Trabalhos Relacionados

O primeiro trabalho a ser comentado é o de [Huang et al. \(2020\)](#), que explorou um novo método de diagnóstico baseado em *Deep Transfer Convolutional Neural Network* (DTCNN) e *Extreme Learning Machine* (ELM), que atuou na classificação de nódulos benignos e malignos. O DTCNN foi adotado pela primeira vez, até então, para extrair características de alto nível de nódulos pulmonares, previamente treinados com o conjunto de dados *ImageNet*. Posteriormente, um classificador ELM foi desenvolvido para classificar nódulos pulmonares benignos e malignos. Os dados utilizados no artigo foram extraídos da base *Lung Image Database Consortium - Image Database Resource Initiative* (LIDC-IDRI) e do conjunto de dados privado *First Affiliated Hospital of Guangzhou Medical University in China* (FAH-GMU). Para o conjunto de dados LIDC-IDRI, os resultados obtidos foram de uma precisão de 94,57%, sensibilidade de 93,69% e especificidade de 95,15%.

No trabalho de [Nishio et al. \(2018\)](#), foi realizada uma comparação entre *Support Vector Machine* (SVM) e *Extreme Gradient Boosting* (XGBoost) como algoritmos de aprendizado de máquina. Os dados utilizados foram imagens de 99 nódulos pulmonares (62 malignos e 37 benignos). O SVM e o XGBoost foram treinados usando o vetor de atributos com seus rótulos correspondentes. O *Tree Parzen Estimator* (TPE) foi utilizado como otimização Bayesiana para parâmetros de SVM e XGBoost. A melhor média de acurácia obtida para os algoritmos foi de 85,00% para o SVM e 89,60% para o XGBoost, que obteve os melhores resultados.

Uma comparação entre os algoritmos MLP, SVM e KNN foi realizada por [Rodrigues et al. \(2018\)](#), utilizando o conjunto de dados LIDC-IDRI, que fornece informações sobre as posições dos nódulos e seus níveis de severidade. Os algoritmos citados foram aplicados a duas tarefas: (1) classificar os nódulos das imagens em malignos ou benignos e (2) classificar os nódulos pulmonares em níveis de severidade (1-5). A abordagem utilizada resultou em uma acurácia de 95,40% para o MLP, 96,70% para o SVM e 95,30% para o KNN.

No artigo de [Poreva et. al \(2017\)](#), uma técnica diferente das demais foi utilizada para classificação, ao invés de imagens de tomografias pulmonares, os classificadores gerados consideraram os métodos básicos de aprendizado de máquina para aplicá-los à tarefa de classificação dos sons pulmonares. Vários parâmetros de sinal foram obtidos com base no conjunto de sons pulmo-

nares. A tarefa do estudo foi classificar sons usando cinco métodos diferentes de aprendizado de máquina. Os resultados mostraram que os métodos de aprendizado de máquina SVM e Árvore de Decisão apresentaram a melhor acurácia, 75% e 72%, respectivamente.

Tran et al. (2019) propuseram um método de aprendizado profundo para melhorar a precisão da classificação de nódulos pulmonares em exames de TC. A arquitetura escolhida foi a de Rede Neural Convolucional Profunda 2D, que possui 15 camadas para extração automática de recursos e classificação das imagens pulmonares como contendo um nódulo ou não. A bases de dados utilizada para geração do classificador foi a LIDC-IDRI. Os resultados obtidos foram de uma acurácia de 97,2%, sensibilidade de 96,0% e especificidade de 97,3%.

Por fim, Dou et al. (2016) propuseram o método da utilização de CNNs tridimensionais (3D) para redução de falsos positivos na detecção automática de nódulos pulmonares a partir de tomografia computadorizada volumétrica. Em comparação com as redes 2D, as CNNs 3D podem codificar informações espaciais mais ricas e extrair recursos mais representativos por meio de sua arquitetura hierárquica treinada com amostras 3D. O conjunto de dados utilizado também foi o LIDC-IDRI. Os resultados obtidos foram uma sensibilidade de 87,00% e especificidade de 99,10%.

3.2 Resumo

Na Tabela 2 é apresentado um resumo dos trabalhos relacionados contendo as referências, metodologia utilizada pelos autores e os resultados mais importantes obtidos em cada um dos trabalhos.

Tabela 2 – Resumo da Revisão Bibliográfica

Referência	Metodologia	Resultados
(HUANG et al., 2020)	ELM & DTCNN	Acurácia - 94,57%
(NISHIO et al., 2018)	SVM & XGBoost	Acurácia: SVM - 85,00% XGBoost - 89,60%
(RODRIGUES et al., 2018)	MLP, SVM, KNN	Acurácia: MLP - 95,40% SVM - 96,70% KNN - 95,30%
(Poreva et. al, 2017)	DT & SVM	Acurácia: DT - 72,00% SVM - 75,00%
(TRAN et al., 2019)	2D CNN	Acurácia - 97,20% Sensibilidade - 96,00% Especificidade - 97,30%
(DOU et al., 2016)	3D CNN	Sensibilidade - 87,00% Especificidade - 99,10%

3.3 Conclusões

A partir dos trabalhos já realizados abordando o tema desse estudo, podemos ver que a utilização de aprendizado profundo no processamento de imagens tem muito potencial, devido às altas acurácias obtidas pelos autores previamente citados. Porém, podemos notar que até o momento não foi realizado um comparativo de redes neurais convolucionais clássicas envolvendo um conjunto de dados binário e um conjunto de dados multiclasse, proposta central do presente estudo.

4 Proposta de Trabalho

Neste capítulo são expostos detalhes a respeito das bases de dados utilizadas no desenvolvimento do trabalho, os softwares e bibliotecas necessários, além da metodologia adotada.

4.1 Bases de Dados

Pretende-se explorar duas bases de dados, sendo uma delas binária e outra multiclasse, para que possamos comparar a execução dos modelos para diferentes aplicações. A seguir detalharemos a quantidade de imagens de cada conjunto de dados, quantidade de pacientes envolvidos, classes existentes e imagens de exemplo.

4.1.1 SPIE-AAPM-NCI Lung Nodule Classification

Este conjunto de dados utilizado é parte de um desafio que foi realizado na Conferência *SPIE Medical Imaging* em 2015, com o apoio da *American Association of Physicists in Medicine* (AAPM) e do *National Cancer Institute* (NCI). O objetivo do desafio era desenvolver e comparar métodos de análise quantitativa de imagens para a classificação diagnóstica de nódulos pulmonares malignos e benignos.

O conjunto de dados é composto por mais de 20.000 imagens de tomografia computadorizada da região torácica, com foco específico nos pulmões contendo tumores. Essas imagens foram coletadas de um grupo de 70 pacientes ao longo de um intervalo de tempo determinado, visando mapear a evolução dos tumores. O conjunto de dados é dividido em duas classes distintas: tumores malignos e tumores benignos. Essa divisão permite a análise e o estudo comparativo entre as características e os padrões visuais presentes nas imagens de ambos os tipos de tumores.

Abaixo, na Tabela 3, temos alguns números importantes a respeito deste conjunto de dados:

Tabela 3 – Detalhes do conjunto de dados SPIE-AAPM-NCI

Modalidades	Tomografia Computadorizada
Qtd de Participantes	70
Qtd de Estudos	70
Qtd de Séries	70
Qtd de Imagens	22.484
Tamanho das Imagens (GB)	12.1

A Figura 19 traz um exemplo de imagem deste conjunto de dados.

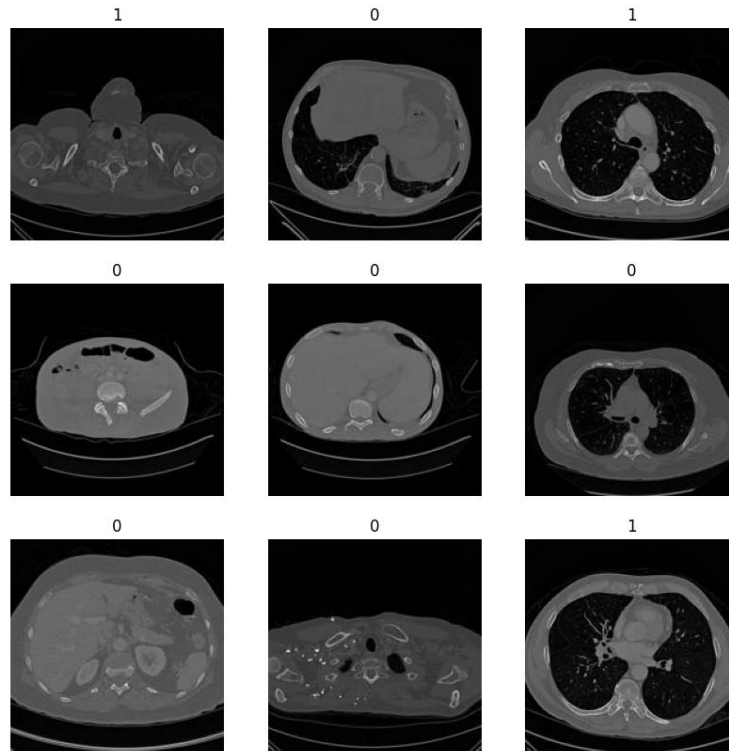


Figura 19 – Exemplos de imagens do conjunto de dados SPIE-AAPM-NCI.
0 - Malignos e 1 - Benignos

Fonte: O Autor

4.1.2 IQ-OTH/NCCD

O conjunto de dados *Iraq-Oncology Teaching Hospital/National Center for Cancer Diseases* (IQ-OTH/NCCD) foi coletado em hospitais iraquianos durante um período de três meses no outono de 2019. Ele inclui tomografias computadorizadas de pacientes diagnosticados com câncer de pulmão em diferentes estágios, bem como indivíduos saudáveis. O conjunto de dados contém um total de 1.097 imagens representando fatias de tomografias computadorizadas de 110 casos. Esses casos são agrupados em três classes: normal, benigno e maligno.

Destes, 40 casos são diagnosticados como malignos; 15 casos com diagnóstico benigno; e 55 casos classificados como casos normais. As tomografias são originalmente coletadas no formato DICOM. Os 110 casos variam em gênero, idade, escolaridade, área de residência e situação de moradia. Alguns deles são funcionários dos ministérios iraquianos dos transportes e do petróleo, outros são agricultores. A maioria deles vem de lugares da região central do Iraque, particularmente das províncias de Bagdá, Wasit, Diyala, Salahuddin e Babilônia. Essa base de dados pode ser acessado online no Kaggle ([ALYASRIY HAMDALLA; AL-HUSEINY, 2020](#)).

A Figura 20 traz alguns exemplos de imagens deste conjunto de dados.

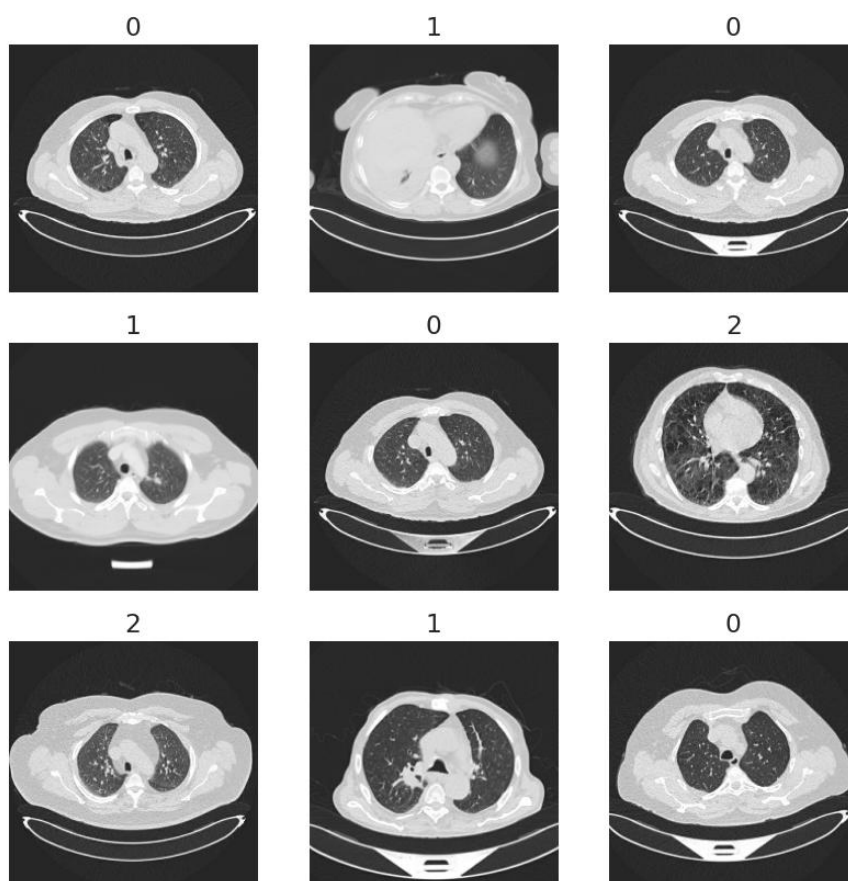


Figura 20 – Exemplos de imagens do conjunto de dados IQ-OTH/NCCD.
0 - Benigno, 1 - Maligno e 2 - Saudável

Fonte: O Autor

4.2 Softwares e Bibliotecas

Para a execução das redes neurais convolucionais aplicadas ao processamento de imagens, faz-se necessário a utilização de determinados softwares e bibliotecas.

A linguagem de programação escolhida para o desenvolvimento do projeto foi o Python (ROSSUM; DRAKE, 2003), que possui estrutura de dados de alto nível e sintaxe elegante. Um dos motivos para a escolha foi o fato do interpretador da linguagem e sua extensa biblioteca estarem disponíveis gratuitamente na web. Além disso, o Python possui inúmeras bibliotecas voltadas à Inteligência Artificial, com muitos algoritmos de aprendizado de máquina já implementados.

Quatro bibliotecas que são extremamente importantes no desenvolvimento do trabalho são: Keras¹, Tensorflow², Matplotlib³ e Scikit-learn⁴. As bibliotecas Keras e TensorFlow são utilizadas para conseguirmos extrair as características e decifrarmos padrões nas imagens. A scikit-

¹ <https://www.tensorflow.org/guide/keras>

² <https://www.tensorflow.org/learn>

³ <https://matplotlib.org/>

⁴ <https://scikit-learn.org/stable/>

learn é uma biblioteca de aprendizado de máquina em python, é através dela que conseguiremos executar as redes neurais convolucionais, que já estão implementadas. Sendo assim, resta agora adequarmos os pesos da rede para o contexto em que estamos inseridos. A matplotlib será muito útil na análise dos resultados, pelo fato de se tratar de uma biblioteca de criação de gráficos.

4.3 Método e Análise da Proposta

Para obter os resultados acerca da problemática exposta, foi realizada uma análise qualitativa da execução de quatro modelos de aprendizado de máquina no processamento de imagens baseados nas arquiteturas VGG16, VGG19, Resnet e Xception. As variáveis que serviram como base para análise foram: acurácia, precisão, *recall*, perda e *f1 score*.

Utilizamos a linguagem de programação Python para os testes, como citado na seção anterior, pelo fato dela possuir bibliotecas prontas com muitos modelos de aprendizado de máquina pré-treinados. Outra ferramenta utilizada nas análises foi o Google Colab, uma plataforma gratuita baseada em nuvem fornecida pelo Google para desenvolvimento e execução de código em Python. Ele oferece um ambiente de notebook interativo que permite aos usuários escrever, executar e compartilhar código, além de fornecer acesso a recursos computacionais, como CPU, GPU e RAM, sem a necessidade de configurar um ambiente local.

Ao final, os dados obtidos foram transformados em gráficos e infográficos para uma melhor visualização, e posteriormente foi feita uma análise comparativa de todos os modelos utilizados, onde conseguimos chegar à conclusão sobre qual é o ideal para o objetivo principal do trabalho, auxiliar profissionais de saúde no diagnóstico do câncer de pulmão.

Após a execução dos modelos aplicados a base de dados binária, selecionamos àqueles que obtiveram os melhores resultados e aplicamos a base de dados multiclasse, para assim avaliarmos se as redes neurais convolucionais conseguiram manter os bons resultados para diferentes abordagens e conjunto de dados.

5 Resultados

Neste capítulo são apresentados os resultados da análise das redes neurais convolucionais selecionadas aplicadas às duas bases de dados citadas.

5.1 *Dataset* Binário: SPIE-AAPM-NCI

Nesta seção será detalhada a execução dos modelos VGG16, VGG19, Resnet e Xception na base SPIE-AAPM-NCI, que possui duas classes: tumores malignos e tumores benignos.

5.1.1 Tratamento das Imagens

As imagens do conjunto de dados *SPIE-AAPM-NCI* estavam no formato DICOM (*Digital Imaging and Communications in Medicine*), um padrão amplamente utilizado na área médica para armazenar, transmitir e compartilhar imagens médicas, como radiografias, tomografias computadorizadas (TC), ressonâncias magnéticas (RM) e ultrassonografias. Para que fosse possível utilizarmos as imagens nos modelos de redes neurais convolucionais, foi necessário a conversão da imagem para o formato *jpg*, processo que foi realizado utilizando Python.

As imagens também possuíam um tamanho de 514×514 pixels, o que inviabilizava o treinamento dos modelos, já que foge ao padrão de tamanho recomendado, além de deixar o treinamento do modelo mais lento. Sendo assim, ainda na etapa de tratamento, redimensionamos as imagens para um tamanho de 180×180 pixels, como mostrado na Figura 21.

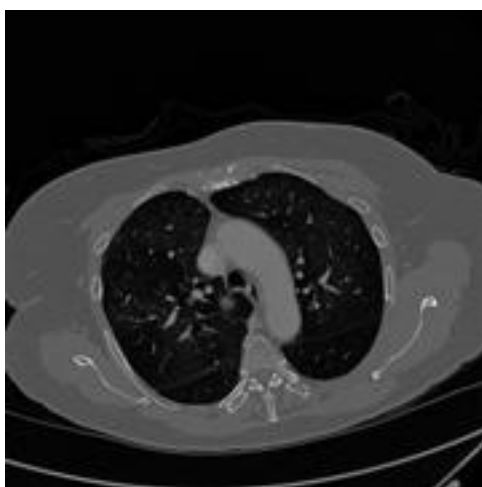


Figura 21 – Exemplo de imagem redimensionada com 180x180 pixels.

Fonte: O Autor

Após a transformação das imagens para o formato e tamanho adequado, realizamos o processo de aumento de dados, uma técnica usada no treinamento de modelos de aprendizado de máquina que consiste em aplicar transformações aleatórias nas imagens do conjunto de dados, a fim de aumentar a diversidade e quantidade de exemplos disponíveis para o modelo aprender. Foram aplicadas várias técnicas para aumentar a diversidade das imagens e melhorar a precisão dos modelos. Entre essas técnicas, destacam-se:

- *Flip*: Essa camada aplica um *flip* horizontal aleatório nas imagens. Isso significa que a imagem pode ser espelhada horizontalmente;
- Rotação: Essa camada aplica uma rotação aleatória às imagens dentro de um limite de ângulo de 0,1 radianos. Isso permite que as imagens sejam rotacionadas de forma aleatória, o que pode ajudar a tornar o modelo mais robusto a diferentes orientações de objetos nas imagens;
- Contraste: Essa camada aplica um ajuste aleatório de contraste nas imagens dentro de um intervalo de $\pm 0,2$. Isso significa que o contraste das imagens pode ser aumentado ou diminuído aleatoriamente, o que pode ajudar a tornar o modelo mais resistente a variações na iluminação das imagens;
- Zoom: Essa camada aplica um zoom aleatório nas imagens dentro de um intervalo de $\pm 0,2$. Isso permite que as imagens sejam ampliadas ou reduzidas aleatoriamente, o que pode ajudar a aumentar a variabilidade dos dados de treinamento e permitir que o modelo seja mais robusto a diferentes tamanhos de objetos nas imagens;
- Translação: Essa camada aplica uma translação aleatória nas imagens dentro de um intervalo de $\pm 0,1$ em ambas as direções horizontal e vertical. Isso significa que as imagens podem ser deslocadas aleatoriamente para cima, para baixo, para a esquerda ou para a direita, o que pode ajudar a tornar o modelo mais resistente a diferentes posições dos objetos nas imagens.

O objetivo dessa etapa no tratamento das imagens foi melhorar a capacidade do modelo em generalizar para novos exemplos, reduzindo o *overfitting* e aumentando a robustez do modelo frente a variações e ruídos presentes nos dados reais. A Figura 22 exibe as variações geradas para uma única imagem do conjunto de dados após a aplicação desse processo.

Concluídas as etapas de tratamento das imagens, prosseguimos para a execução dos modelos.

5.1.2 Execução dos Modelos

O conjunto de dados foi dividido em treino (75%) e teste (25%), totalizando 15.268 imagens para treino e 3.816 imagens para teste. Vale ressaltar que nessa divisão foi mantido o equilí-

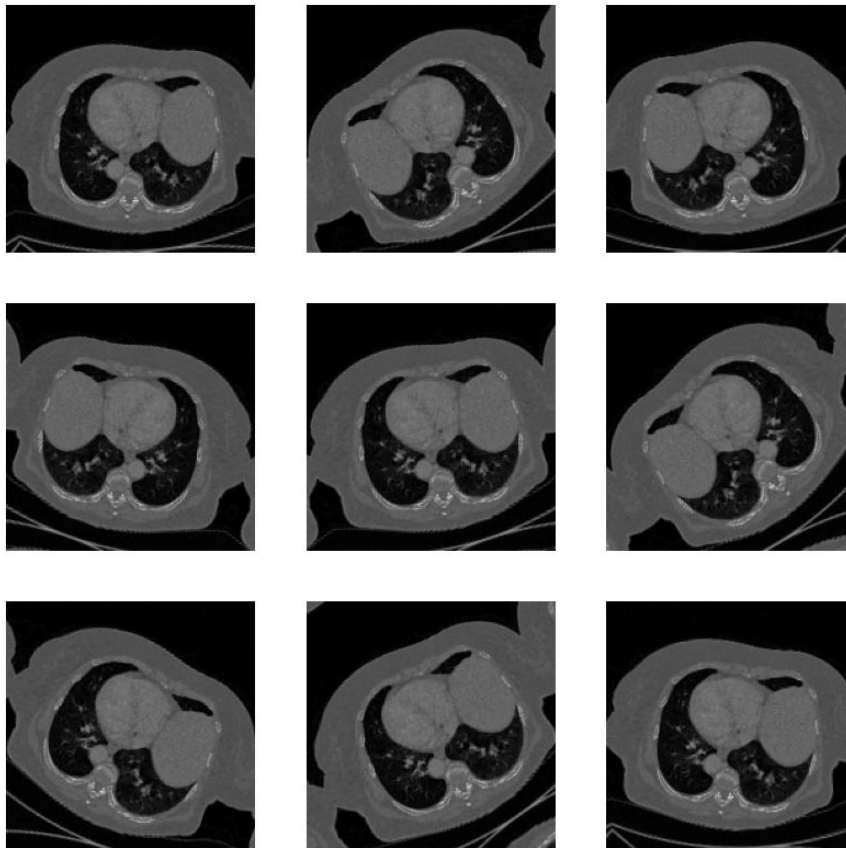


Figura 22 – Processo de aumento de dados.

Fonte: O Autor

brio entre as duas classes para cada um dos subconjuntos.

Na etapa de divisão do conjunto de dados, foi utilizado uma semente (*seed*) de 1337. Esse parâmetro é importante pois controla a aleatoriedade em certas operações, e em nosso caso, garantia que a divisão dos dados entre o conjunto de treinamento, teste e validação fosse sempre a mesma, mesmo que o código fosse executado várias vezes. Dessa forma, poderíamos focar apenas nos ajustes dos parâmetros do modelo na fase de treinamento, sem se preocupar com alterações advindas da mudança da divisão do conjunto de imagens.

Feita a divisão das imagens, construímos os modelos baseados nas arquiteturas de redes neurais convolucionais já citadas: VGG16, VGG19, Resnet e Xception. Carregamos o modelo base pré-treinado no conjunto de dados *ImageNet* e adicionamos a esse modelo base camadas que incluem redimensionamento, convolução, normalização em lotes, ativação, convolução separável, pooling global médio e dropout. Ao final, definimos a camada de saída do modelo, que possui uma única unidade com função de ativação sigmoideal. Isso significa que o modelo é usado para uma tarefa de classificação binária, onde a saída é uma probabilidade entre 0 e 1.

Para todos os 4 modelos analisados, foram executadas 70 épocas utilizando o otimizador *Adam* a uma taxa de aprendizagem de 1×10^{-5} . A função de perda escolhida foi a *binary*

crossentropy, essa função é amplamente utilizada em problemas de classificação binária, especialmente quando se trabalha com modelos de aprendizado profundo. Ela calcula a diferença entre as probabilidades previstas pelo modelo e as classes reais dos dados de treinamento. Essa função mede a discrepância entre a distribuição de probabilidade prevista e a verdadeira distribuição, atribuindo uma penalidade maior quando as previsões estão longe do valor real. Isso ajuda o modelo a aprender a discriminar corretamente entre as duas classes e a fazer previsões mais precisas.

5.1.2.1 VGG16

Na Figura 23 podemos ver como ficou a arquitetura da rede baseada no modelo VGG16. Já na Tabela 4 podemos visualizar um resumo com os resultados do modelo. Podemos também entender como foi a evolução do treinamento da rede observando os gráficos de perda, acurácia, precisão, *recall* e *f1 score* da Figura 24.

Tabela 4 – *Dataset* Binário - Resumo dos resultados do modelo VGG16

Acurácia	0,96147
Perda	0,11648
Precisão	0,92701
<i>F1 Score</i>	0,96196
<i>Recall</i>	1,0

5.1.2.2 VGG19

Na Figura 25 podemos ver como ficou a arquitetura da rede baseada no modelo VGG19. Já na Tabela 5 podemos visualizar um resumo com os resultados do modelo. Podemos também entender como foi a evolução do treinamento da rede observando os gráficos de perda, acurácia, precisão, *recall* e *f1 score* da Figura 26.

Tabela 5 – *Dataset* Binário - Resumo dos resultados do modelo VGG19

Acurácia	0,99371
Perda	0,02032
Precisão	0,99569
<i>F1 Score</i>	0,99343
<i>Recall</i>	0,99143

5.1.2.3 Resnet

Na Figura 27 podemos ver como ficou a arquitetura da rede baseada no modelo Resnet. Já na Tabela 6 podemos visualizar um resumo com os resultados do modelo. Podemos também entender como foi a evolução do treinamento da rede observando os gráficos de perda, acurácia, precisão, *recall* e *f1 score* da Figura 28.

Tabela 6 – *Dataset* Binário - Resumo dos resultados do modelo Resnet

Acurácia	0,99109
Perda	0,02350
Precisão	0,98211
<i>F1 Score</i>	0,99106
<i>Recall</i>	1,0

5.1.2.4 Xception

Na Figura 29 podemos ver como ficou a arquitetura da rede baseada no modelo Xception. Já na Tabela 7 podemos visualizar um resumo com os resultados do modelo. Podemos também entender como foi a evolução do treinamento da rede observando os gráficos de perda, acurácia, precisão, *recall* e *f1 score* da Figura 30.

Tabela 7 – *Dataset* Binário - Resumo dos resultados do modelo Xception

Acurácia	0,99868
Perda	0,00429
Precisão	0,99786
<i>F1 Score</i>	0,99856
<i>Recall</i>	0,99946

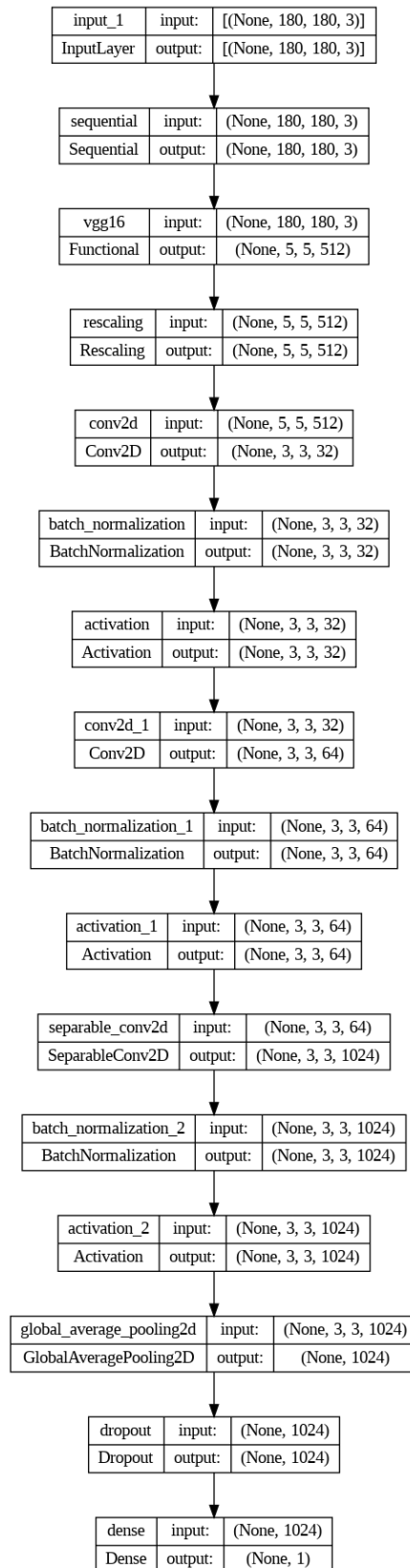


Figura 23 – Dataset Binário - Modelo VGG16.

Fonte: O Autor

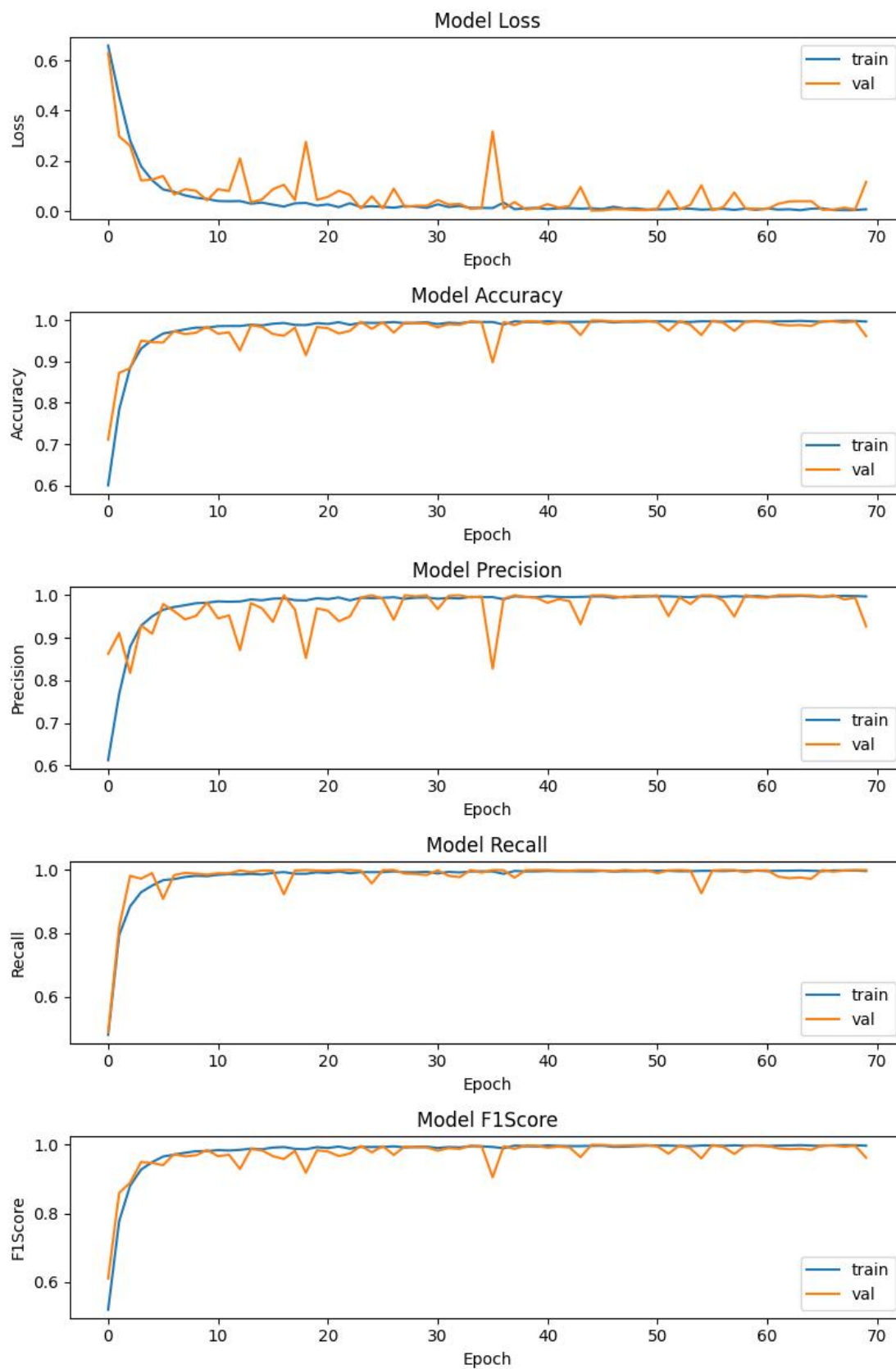


Figura 24 – Dataset Binário - Resultados VGG16.

Fonte: O Autor

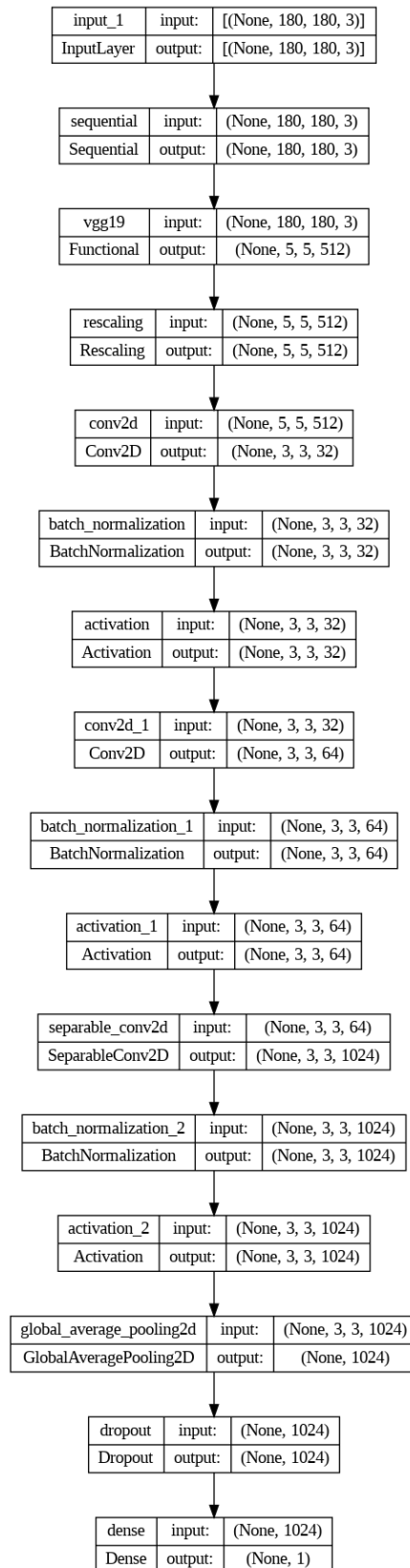


Figura 25 – Dataset Binário - Modelo VGG19.

Fonte: O Autor

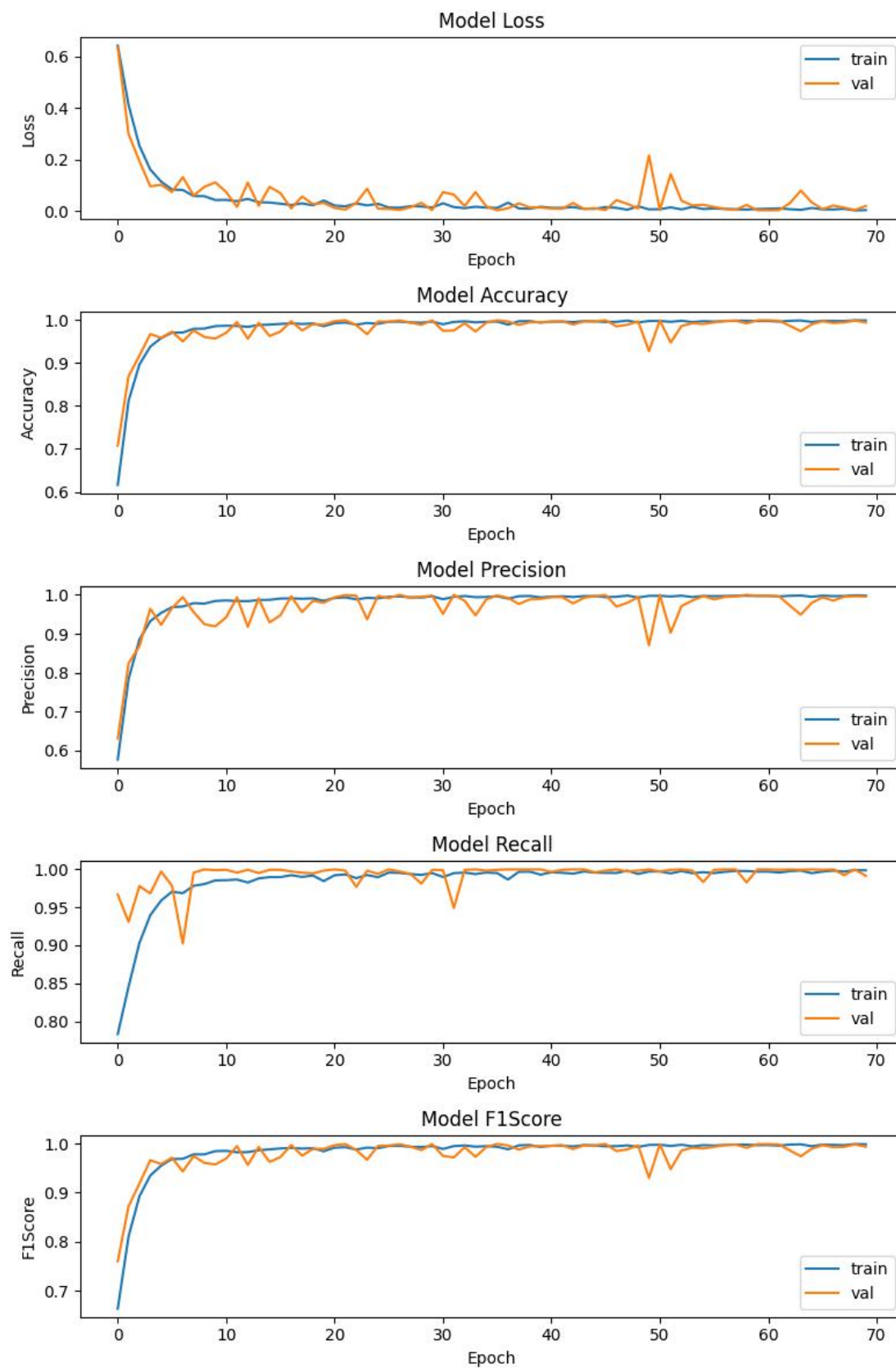


Figura 26 – Dataset Binário - Resultados VGG19.

Fonte: O Autor

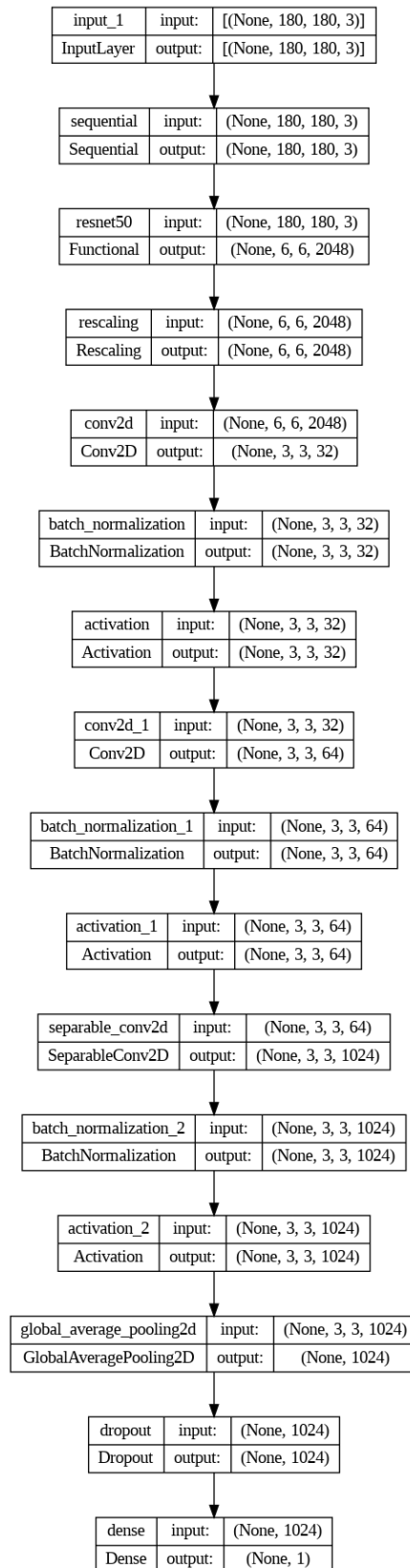


Figura 27 – Dataset Binário - Modelo Resnet.

Fonte: O Autor

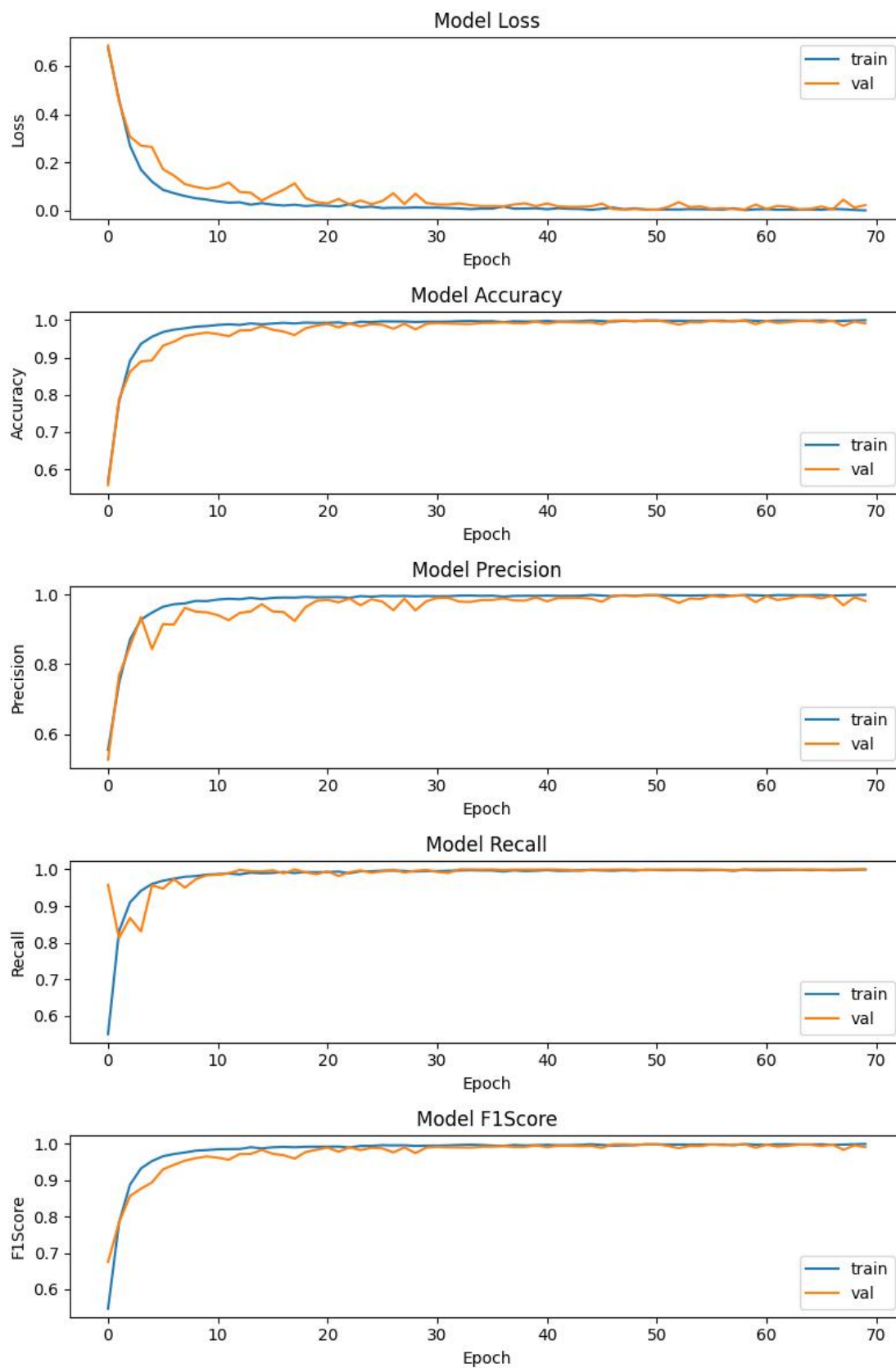


Figura 28 – Dataset Binário - Resultados Resnet.

Fonte: O Autor

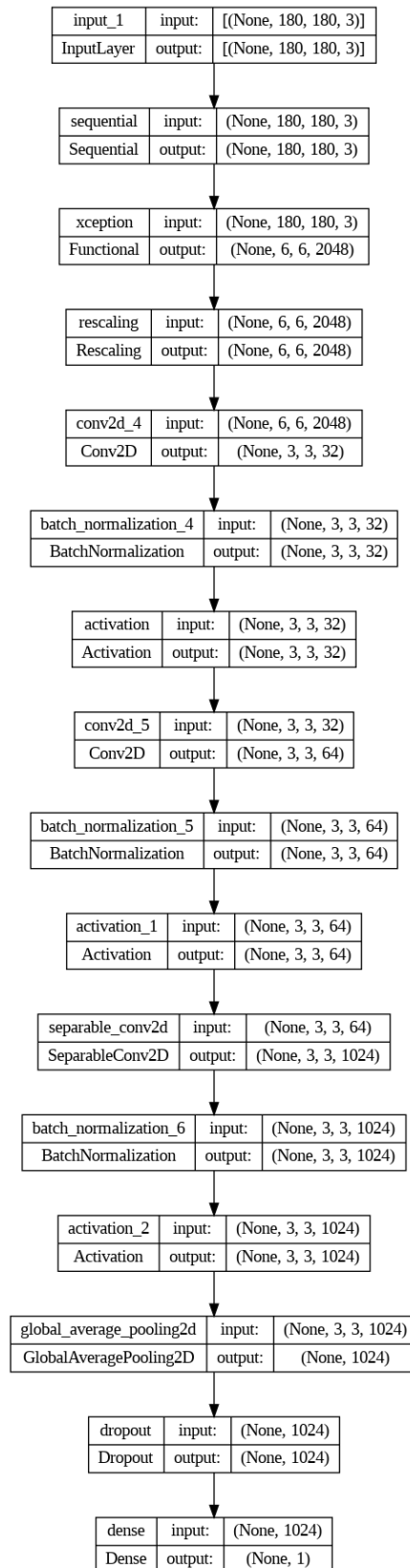


Figura 29 – Dataset Binário - Modelo Xception.

Fonte: O Autor

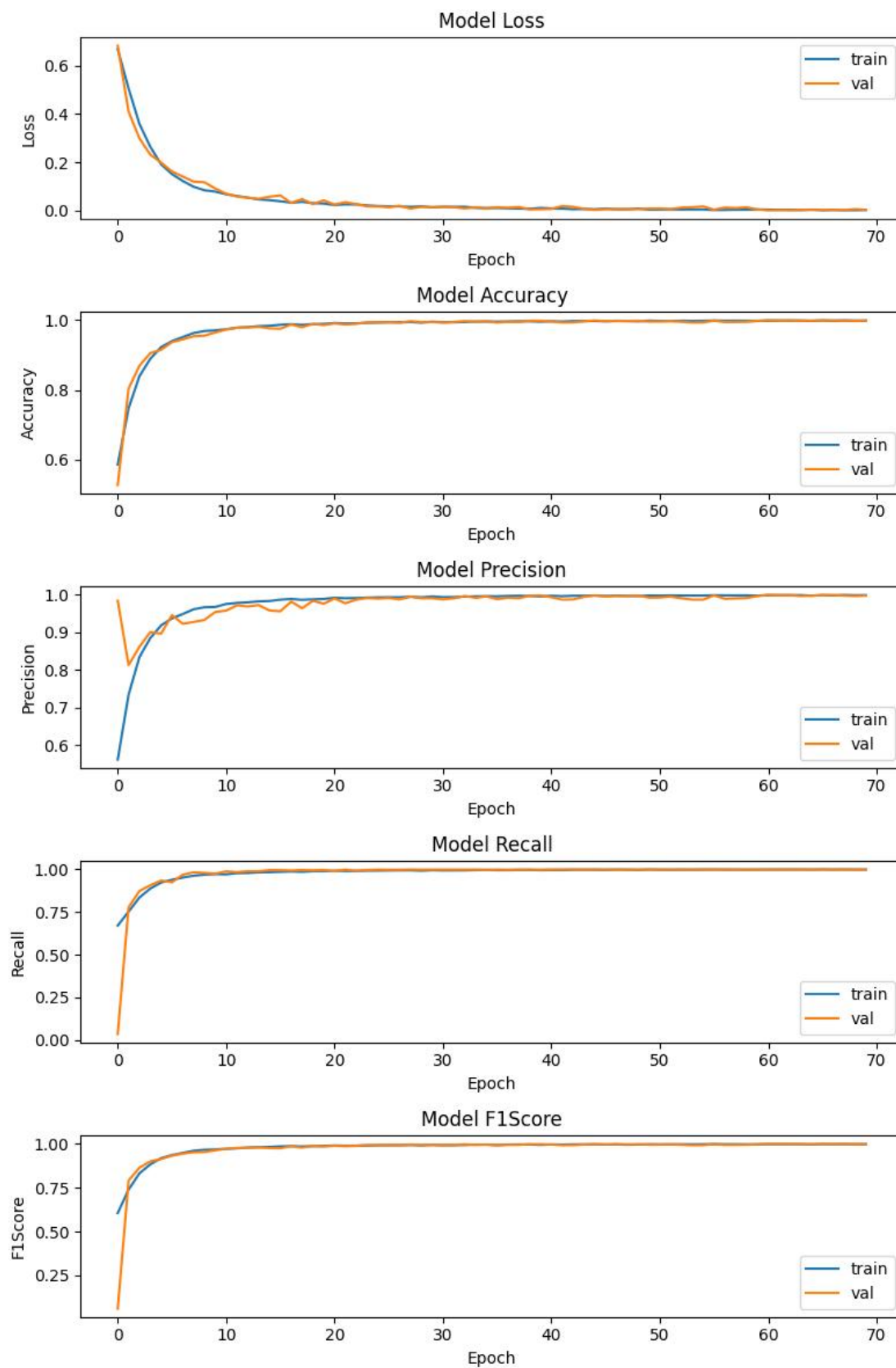


Figura 30 – Dataset Binário - Resultados Xception.

Fonte: O Autor

5.1.3 Grad-CAM

Utilizamos a técnica Grad-CAM (2.5.3), para entender quais partes das imagens dos tumores eram mais relevantes para a classificação feita pelos modelos treinados. Foram selecionadas duas imagens do conjunto de dados, uma contendo um tumor maligno e outra contendo um tumor benigno. Abaixo veremos, para cada um dos modelos, um mapa de calor que mostra qual parte da imagem era a principal para fazer a classificação.

5.1.3.1 VGG16

As Figuras 31 e 32 exibem os mapas de calor da rede VGG16 para exemplos de tumores benigno e maligno, respectivamente.

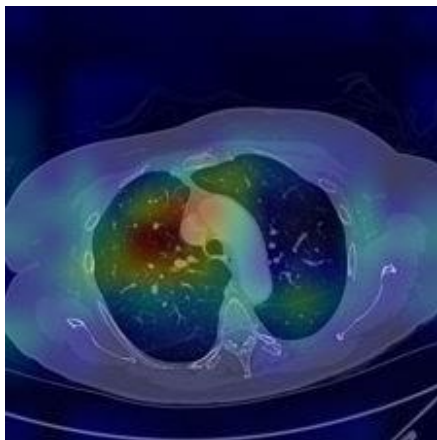


Figura 31 – *Dataset* Binário - VGG16 - Mapa de calor para tumor benigno.

Fonte: O Autor

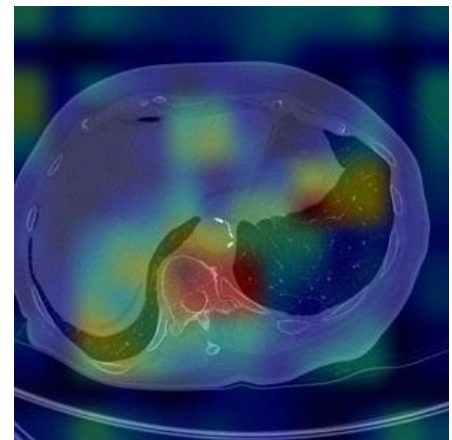


Figura 32 – *Dataset* Binário - VGG16 - Mapa de calor para tumor maligno.

Fonte: O Autor

5.1.3.2 VGG19

As Figuras 33 e 34 exibem os mapas de calor da rede VGG19 para exemplos de tumores benigno e maligno, respectivamente.

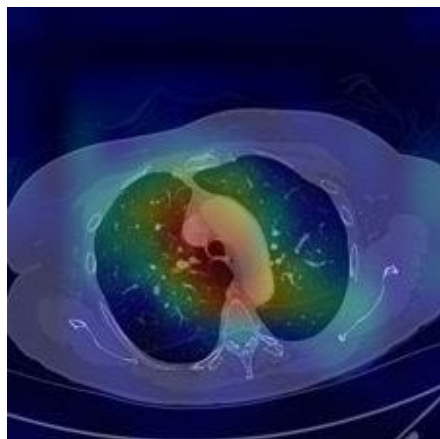


Figura 33 – *Dataset* Binário - VGG19 - Mapa de calor para tumor benigno.

Fonte: O Autor

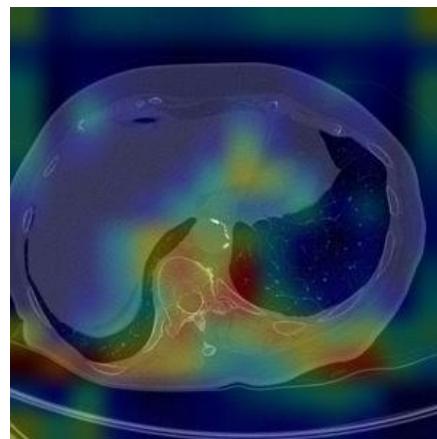


Figura 34 – *Dataset* Binário - VGG19 - Mapa de calor para tumor maligno.

Fonte: O Autor

5.1.3.3 Resnet

As Figuras 35 e 36 exibem os mapas de calor da rede Resnet para exemplos de tumores benigno e maligno, respectivamente.

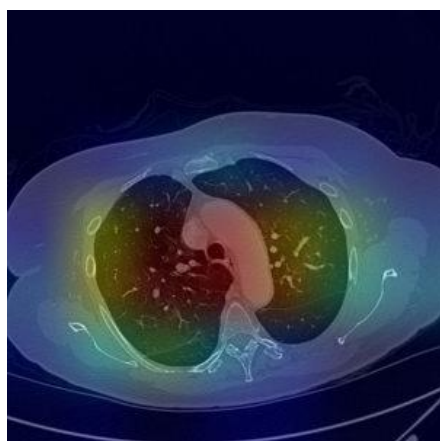


Figura 35 – *Dataset* Binário - Resnet - Mapa de calor para tumor benigno.

Fonte: O Autor

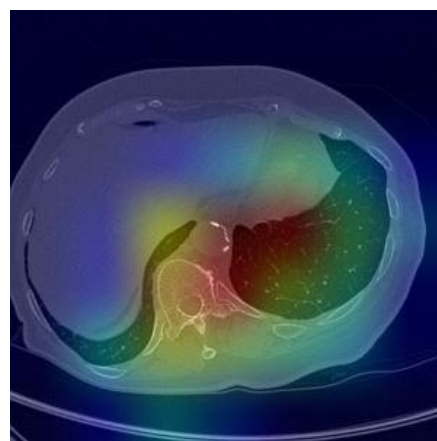


Figura 36 – *Dataset* Binário - Resnet - Mapa de calor para tumor maligno.

Fonte: O Autor

5.1.3.4 Xception

As Figuras 37 e 38 exibem os mapas de calor da rede Xception para exemplos de tumores benigno e maligno, respectivamente.

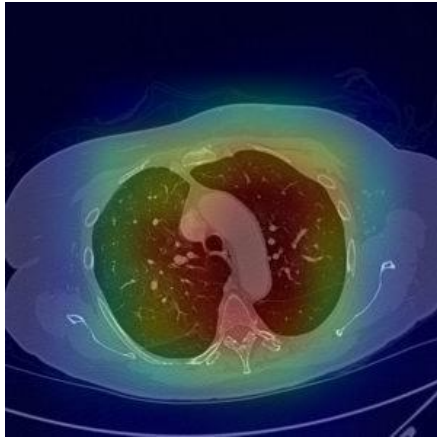


Figura 37 – *Dataset* Binário - Xception - Mapa de calor para tumor benigno.

Fonte: O Autor

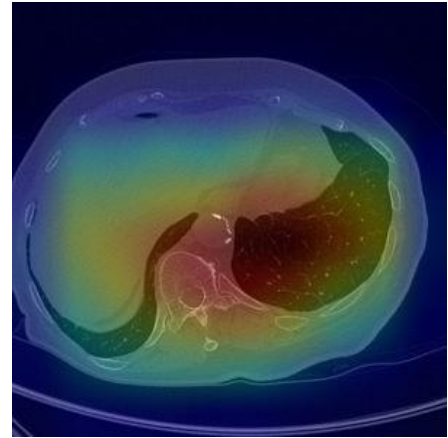


Figura 38 – *Dataset* Binário - Xception - Mapa de calor para tumor benigno.

Fonte: O Autor

5.1.4 Parâmetros das Redes

Os parâmetros de uma rede neural convolucional desempenham um papel crucial no processo de aprendizado da rede. Existem três principais tipos de parâmetros em uma CNN: pesos (*weights*), vies (*bias*) e hiperparâmetros (LECUN; BENGIO; HINTON, 2015).

A importância dos parâmetros reside no fato de que eles determinam a capacidade de uma rede neural convolucional em aprender e representar os dados de forma eficaz. O ajuste adequado dos pesos e vieses é fundamental para que a rede seja capaz de extrair características relevantes dos dados de entrada e tomar decisões precisas. Além disso, a escolha adequada dos hiperparâmetros pode impactar significativamente o desempenho do modelo, influenciando fatores como a taxa de convergência, a capacidade de generalização e a estabilidade do treinamento.

Não há uma equivalência direta que garanta um melhor resultado em redes com mais parâmetros. Porém, em linhas gerais, uma rede com mais parâmetros pode ter maior capacidade de aprendizado e ser capaz de representar relações mais complexas nos dados, em contraponto, ela também corre o risco de se tornar excessivamente ajustada aos dados de treinamento, levando ao *overfitting*. Portanto, encontrar um equilíbrio adequado entre a complexidade do modelo e a quantidade de parâmetros é fundamental para obter bons resultados (GOODFELLOW; BENGIO; COURVILLE, 2016b).

Na Tabela 8 podemos visualizar qual a quantidade de parâmetros de cada um dos modelos analisados, essa informação será importante para avaliarmos os resultados obtidos.

Tabela 8 – Quantidade de parâmetros dos modelos treinados

Modelo	Parâmetros
VGG16	14.953.313
VGG19	20.263.009
Xception	21.542.473
Resnet	24.268.705

5.1.5 Matriz de Confusão

A matriz de confusão é uma ferramenta essencial na avaliação de modelos de rede neural, pois fornece uma visão detalhada do desempenho do modelo em termos de classificação. Ela mostra a contagem de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos para cada classe de interesse. A matriz de confusão ajuda a identificar se o modelo está tendo dificuldades em distinguir certas classes, detectar falsos positivos/negativos ou se há algum desequilíbrio na distribuição dos dados. A partir dessas informações, é possível tomar medidas corretivas, como ajustar limiares de decisão ou coletar mais dados para classes representadas (FAWCETT, 2006).

Abaixo veremos como ficaram as matrizes de confusão dos modelos treinados e discutiremos o que essas informações significam de fato em nossa análise.

5.1.5.1 VGG16

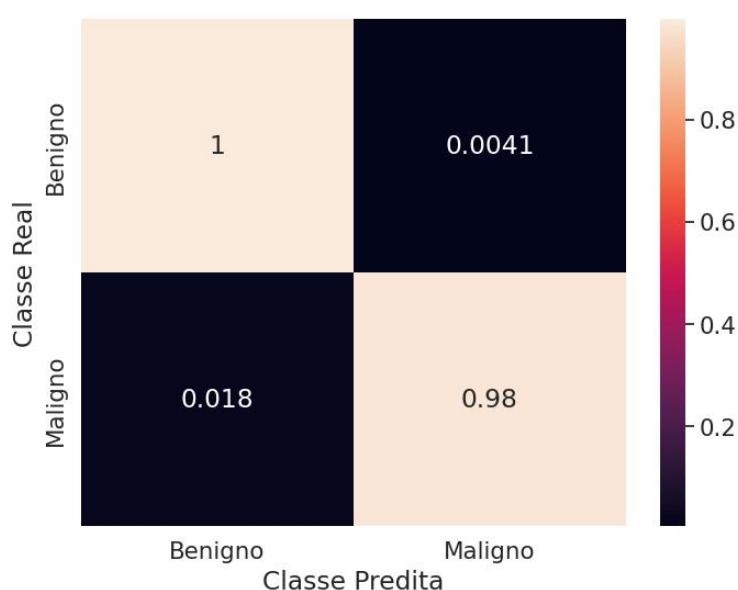
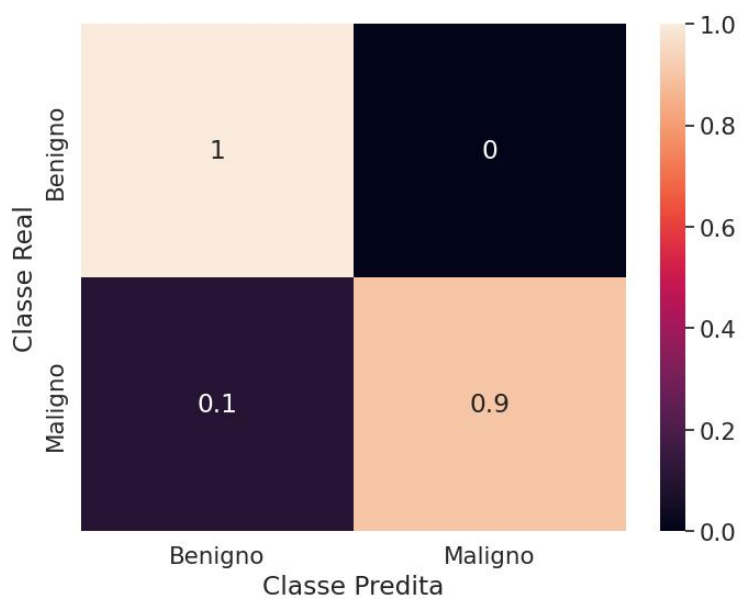


Figura 39 – Dataset Binário - Matriz de Confusão VGG16.

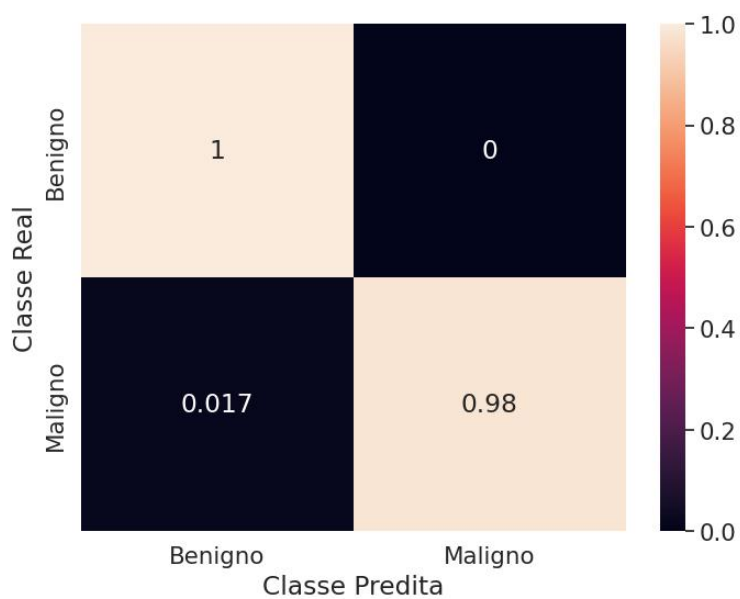
Fonte: O Autor

5.1.5.2 VGG19

Figura 40 – *Dataset* Binário - Matriz de Confusão VGG19.

Fonte: O Autor

5.1.5.3 Resnet

Figura 41 – *Dataset* Binário - Matriz de Confusão Resnet.

Fonte: O Autor

5.1.5.4 Xception

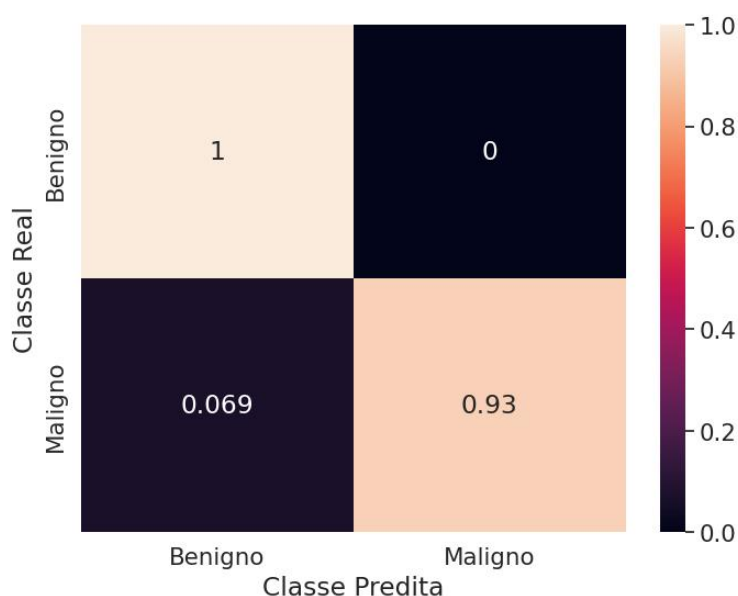


Figura 42 – Dataset Binário - Matriz de Confusão Xception.

Fonte: O Autor

Pelo fato de termos definido a camada de saída do modelo em uma única unidade com função de ativação sigmoideal, o resultado das previsões, dado um conjunto de imagens, será uma probabilidade entre zero e um. Ao plotar a matriz de confusão para os modelos, foi necessário definirmos um limiar de decisão, um valor que determina a fronteira entre as classes de tumores benigno e maligno. O limiar definido foi de 0,999, sendo que os valores menores que esse limiar eram classificados como benigno, e os valores maior que o limiar considerados maligno. Esse processo foi feito para equilibrar a taxa de falsos positivos e falsos negativos.

Ao analisar as matrizes geradas, podemos notar que todos os modelos tiveram mais facilidade para classificar os tumores benignos de maneira assertiva. A rede VGG16, como podemos ver na Figura 39 foi a única que classificou algumas poucas imagens de tumores benignos como sendo malignos. Isso explica o motivo pelo qual o modelo obteve resultados inferiores as demais redes. Avançando para a rede VGG19, como mostrado na Figura 40, vemos uma taxa de assertividade na classificação dos tumores benignos em 100%, porém em relação aos tumores malignos, 10% dos dados do conjunto de teste foram classificados de maneira errônea. Em relação aos testes de matriz de confusão, a rede Resnet obteve os melhores resultados, como podemos visualizar na Figura 41, menos de 1% das imagens do conjunto de validação foram classificadas de maneira errada, tumores que eram malignos, mas foram classificados como benignos. Por fim, a rede Xception, exibida na Figura 42, obteve 7% de erro nas classificações de tumores malignos.

Os resultados expostos estão diretamente ligados a rigidez imposta de maneira intencional do limiar de decisão, caso ajustássemos esse limiar para 0,9%, por exemplo, já teríamos resultados

melhores em relação a falsos positivos e falsos negativos, em especial para a rede Xception, que com esse novo limiar, obteve uma taxa de acerto de 99% para os tumores malignos e 100% para os tumores benignos.

5.1.6 Avaliação dos Resultados

Ao analisar os resultados obtidos e os gráficos de treinamento dos modelos, foi possível notar que todos eles apresentaram uma acurácia acima de 96%, chegando até mesmo em alguns casos a 99%. No entanto, pode-se observar também que os modelos com maior quantidade de parâmetros, como a Resnet e Xception, tiveram uma vantagem em relação aos modelos baseados na arquitetura VGG16 e VGG19. Essa diferença de desempenho pode ser atribuída a vários fatores. Primeiramente, os modelos com mais parâmetros possuem maior capacidade de aprendizado e podem capturar mais detalhes e nuances nos dados, e as redes baseadas na arquitetura VGG possuem menos parâmetros que as demais redes treinadas. Além disso, a arquitetura das redes Resnet e Xception foram projetadas com técnicas avançadas, como conexões residuais e convoluções separáveis, que ajudam a melhorar a capacidade de representação do modelo. Apesar dos modelos com mais parâmetros, em linhas gerais, aumentarem o tempo e os recursos necessários para treiná-los, nos testes realizados, esse fator foi determinante para obtermos melhores resultados.

Ao observar os mapas de calor gerados utilizando a técnica Grad-CAM, notamos que os modelos que obtiveram os melhores resultados foram aqueles que classificavam como parte mais relevante da imagem toda a abrangência dos pulmões, como podemos visualizar no modelo Xception, rede que obteve o melhor resultado de acurácia e precisão, exposto nas Figuras 37 e 38. Ao observarmos as regiões das imagens mais relevantes para as redes VGG16 e VGG19 na classificação, podemos ver que elas não conseguiram abranger a área total do pulmão, deixando assim, em alguns casos, de detectar os tumores corretamente, o que causou um desempenho inferior às demais redes.

5.2 Dataset Multiclasse: IQ-OTH/NCCD

Nesta seção será detalhada a execução dos modelos que obtiveram os melhores resultados para a base de dados binária, Resnet e Xception, aplicadas a base IQ-OTH/NCCD, que possui três classes: tumores malignos, tumores benignos e normal (pulmões sem tumor).

5.2.1 Tratamento das Imagens

Pelo fato desse conjunto de dados possuir apenas 1097 imagens, uma quantidade muito menor frente a primeira base, que possuía aproximadamente 20 mil imagens, elas foram redimensionadas para um tamanho de 224×224 pixels, como mostrado na Figura 43, colaborando assim

para que as redes neurais convolucionais selecionadas, Resnet e Xception, possuam um melhor desempenho.



Figura 43 – Exemplo de imagem redimensionada com 224x224 pixels.

Fonte: O Autor

5.2.2 Execução dos Modelos

O conjunto de dados foi dividido em treino (75%) e teste (25%), totalizando 823 imagens para treino e 274 imagens para teste. Vale ressaltar que nessa divisão foi mantido o equilíbrio entre as três classes para cada um dos subconjuntos.

Feita a divisão das imagens, construímos os modelos baseados nas arquiteturas de redes neurais convolucionais já citadas: Resnet e Xception. Carregamos o modelo base pré-treinado no conjunto de dados *ImageNet* e adicionamos a esse modelo base camadas que incluem redimensionamento, convolução, normalização em lotes, ativação, convolução separável, *pooling* global médio e *dropout*. Ao final, definimos a camada de saída do modelo, que possui três unidades com função de ativação *softmax*. Isso significa que o modelo é utilizado para classificar um conjunto de dados com três classes.

Para os 2 modelos analisados, foram executadas 35 épocas, a metade das épocas utilizadas para a primeira base, já que a quantidade de imagens dessa base era inferior. O otimizador utilizado foi o *Adam* a uma taxa de aprendizagem de 1×10^{-5} . A função de perda escolhida foi a *sparse categorical crossentropy*, utilizada em problemas de classificação multiclasse quando as classes são representadas por rótulos inteiros. Ela é especialmente útil quando as classes não estão codificadas no formato *one-hot* (vetor binário onde apenas uma posição é 1 e as outras são 0), mas sim representadas por valores inteiros distintos.

5.2.2.1 Resnet

Na Figura 44 podemos ver como ficou a arquitetura da rede baseada no modelo Resnet. Já na Tabela 9 podemos visualizar um resumo com os resultados do modelo. Podemos também entender como foi a evolução do treinamento da rede observando os gráficos de perda e acurácia da Figura 45.

Tabela 9 – *Dataset* Multiclasse - Resumo dos resultados do modelo Resnet

Acurácia	0,9927
Perda	0,0920
Precisão	0,9800
<i>F1 Score</i>	0,9900
<i>Recall</i>	0,9900

5.2.2.2 Xception

Na Figura 46 podemos ver como ficou a arquitetura da rede baseada no modelo Xception. Já na Tabela 10 podemos visualizar um resumo com os resultados do modelo. Podemos também entender como foi a evolução do treinamento da rede observando os gráficos de perda e acurácia da Figura 47.

Tabela 10 – *Dataset* Multiclasse - Resumo dos resultados do modelo Xception

Acurácia	0,9927
Perda	0,0688
Precisão	0,9900
<i>F1 Score</i>	0,9900
<i>Recall</i>	0,9900

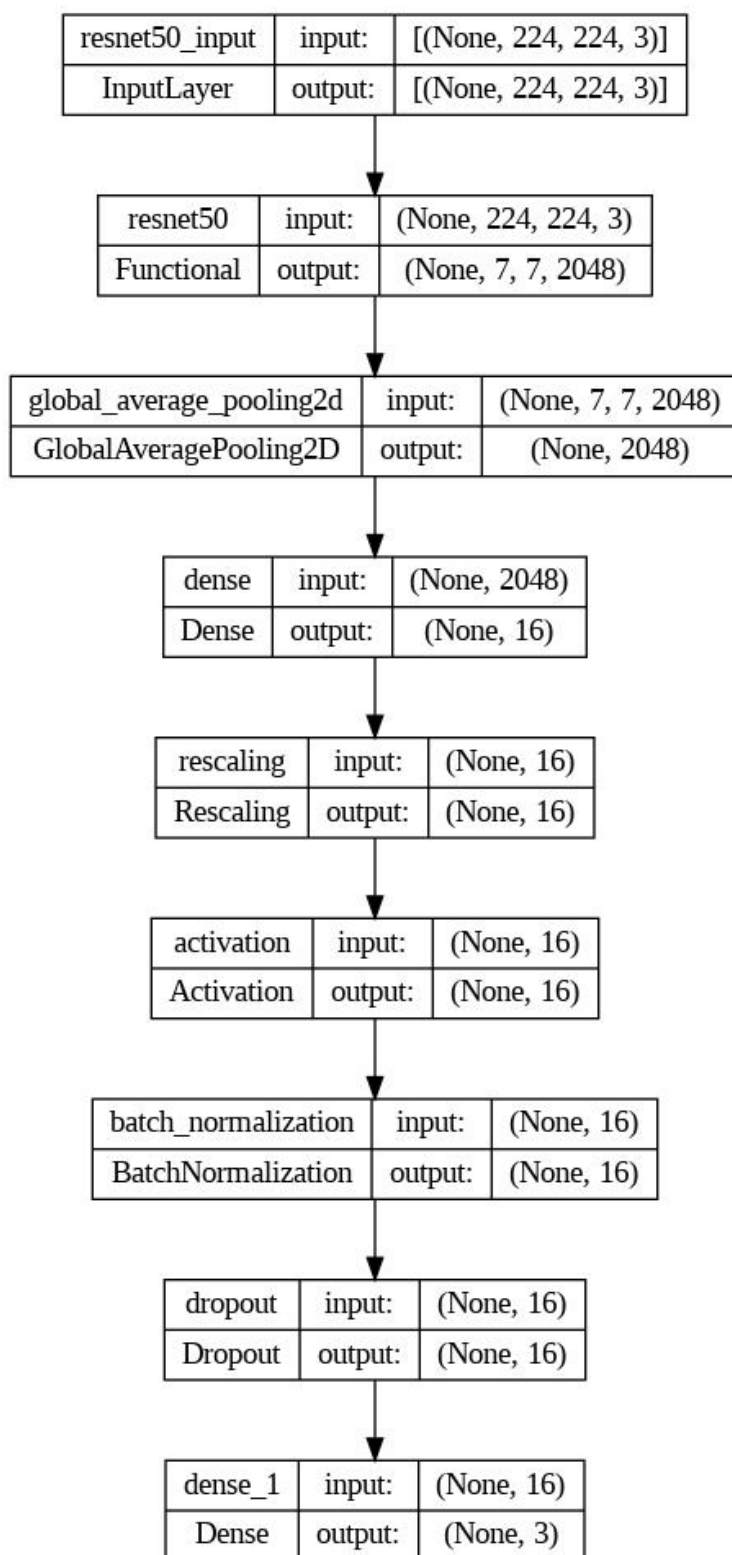


Figura 44 – Dataset Multiclasse - Modelo Resnet.

Fonte: O Autor

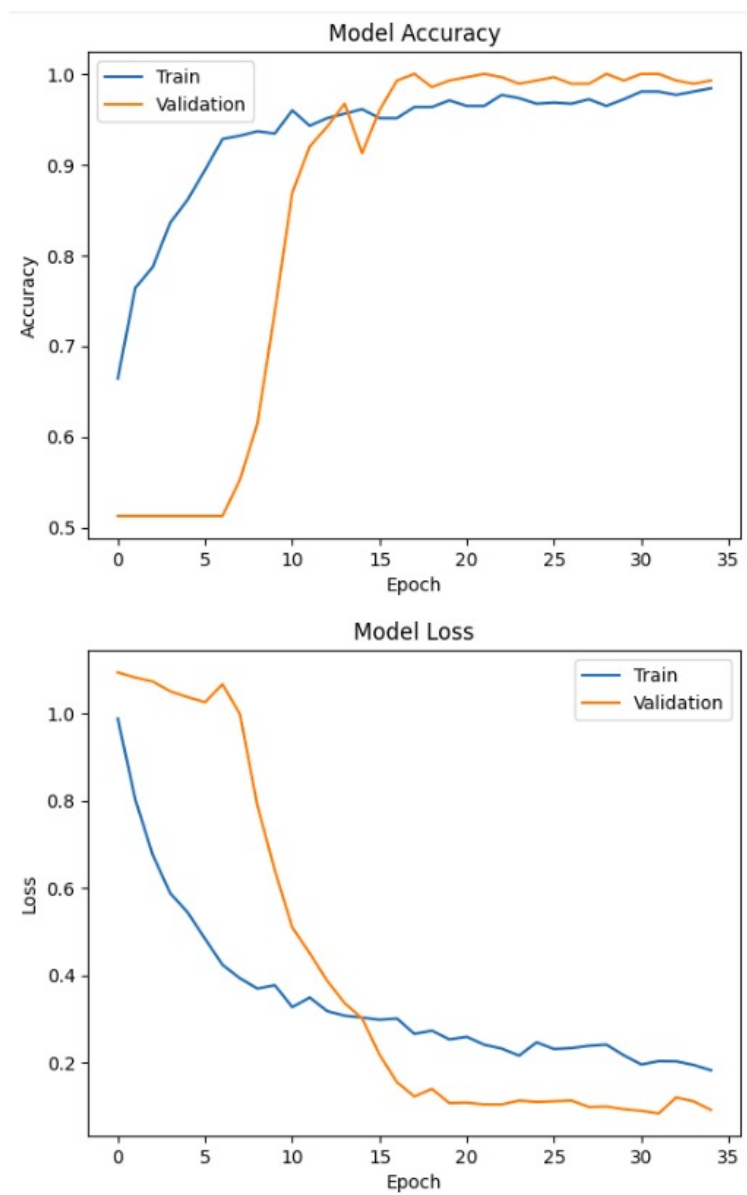


Figura 45 – *Dataset* Multiclasse - Resultados Resnet.

Fonte: O Autor

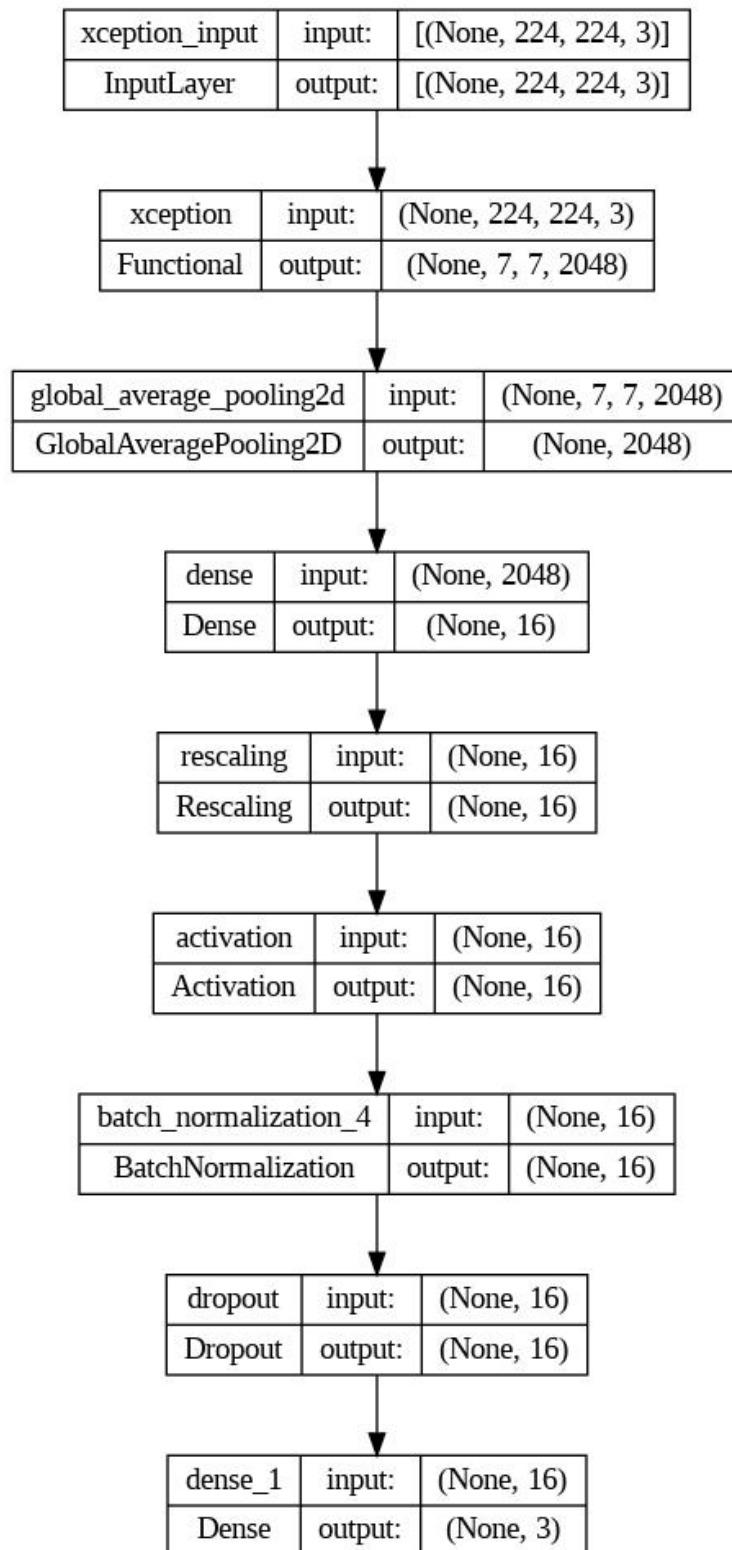


Figura 46 – Dataset Multiclasse - Modelo Xception.

Fonte: O Autor

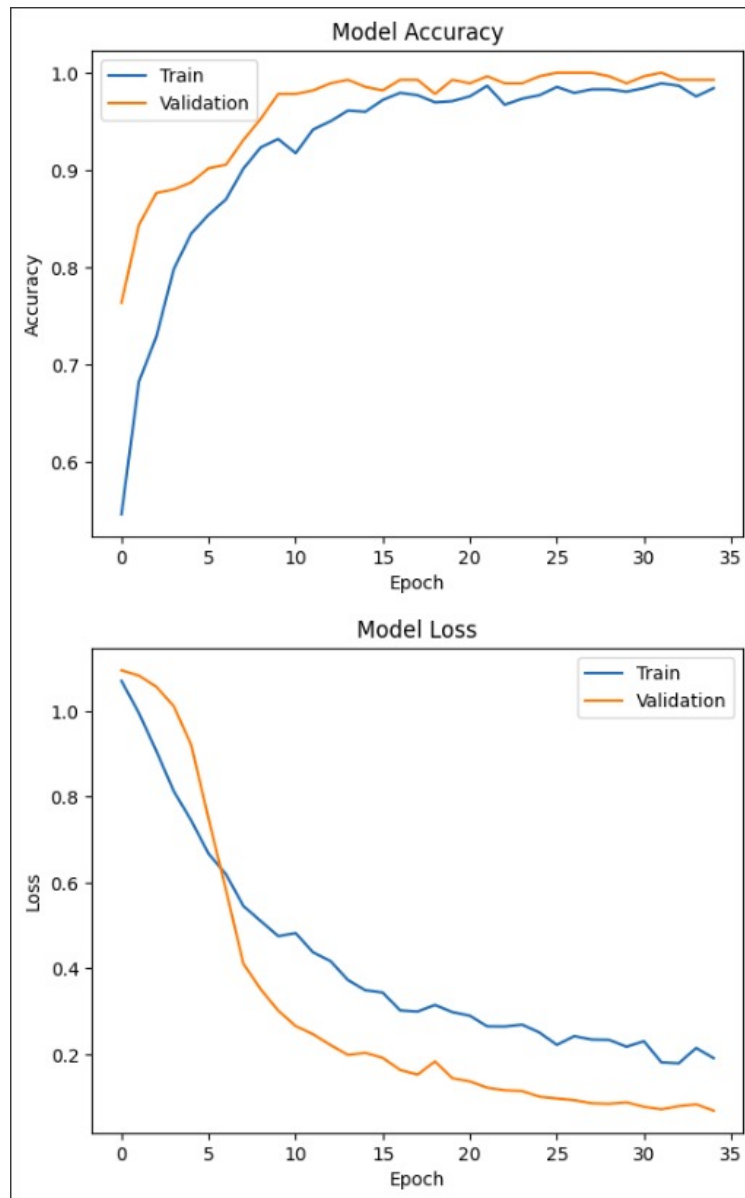


Figura 47 – Dataset Multiclasse - Resultados Xception.

Fonte: O Autor

5.2.3 Grad-CAM

Assim como foi feito para o conjunto de dados binário, também utilizamos a técnica Grad-CAM (2.5.3) no conjunto de dados multiclasse para entender quais partes das imagens dos pulmões eram mais relevantes para a classificação feita pelos modelos treinados. Nesse caso, foram selecionadas três imagens do conjunto de dados, sendo uma de cada classe: benigno, maligno e normal. Abaixo veremos, para cada um dos modelos, um mapa de calor que mostra qual parte da imagem era a principal para fazer a classificação.

5.2.3.1 Resnet

As Figuras 48, 49 e 50 exibem os mapas de calor da rede Resnet para imagens das classes benigno, maligno e normal, respectivamente.



Figura 48 – *Dataset* Multiclasse - Resnet - Mapa de calor para tumor benigno.

Fonte: O Autor

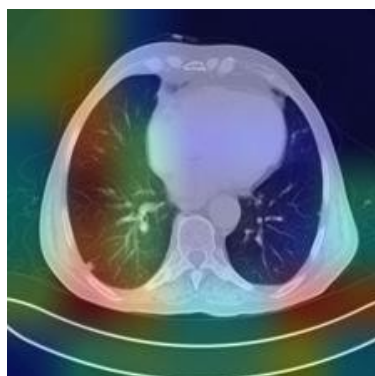


Figura 49 – *Dataset* Multiclasse - Resnet - Mapa de calor para tumor maligno.

Fonte: O Autor



Figura 50 – *Dataset* Multiclasse - Resnet - Mapa de calor para pulmões sem tumor.

Fonte: O Autor

5.2.3.2 Xception

As Figuras 51, 52 e 53 exibem os mapas de calor da rede Xception para imagens das classes benigno, maligno e normal, respectivamente.

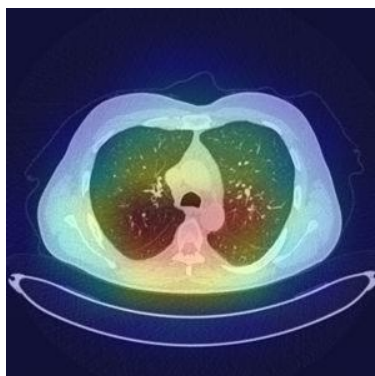


Figura 51 – *Dataset* Multiclasse - Xception - Mapa de calor para tumor benigno.

Fonte: O Autor



Figura 52 – *Dataset* Multiclasse - Xception - Mapa de calor para tumor maligno.

Fonte: O Autor



Figura 53 – *Dataset* Multiclasse - Xception - Mapa de calor para pulmões sem tumor.

Fonte: O Autor

5.2.4 Matriz de Confusão

Abaixo veremos como ficaram as matrizes de confusão dos modelos treinados e discutiremos o que essas informações significam de fato em nossa análise do conjunto de dados multiclasse.

5.2.4.1 Resnet

A Figura 54 exibe a matriz de confusão da rede Resnet.

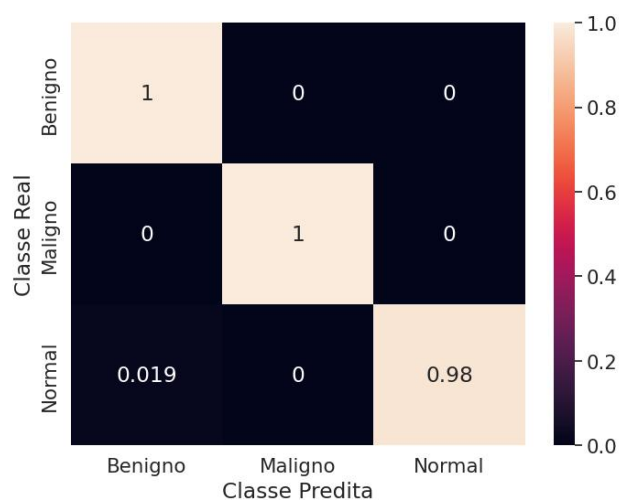


Figura 54 – Dataset Multiclasse - Matriz de Confusão Resnet.

Fonte: O Autor

5.2.4.2 Xception

A Figura 55 exibe a matriz de confusão da rede Xception.

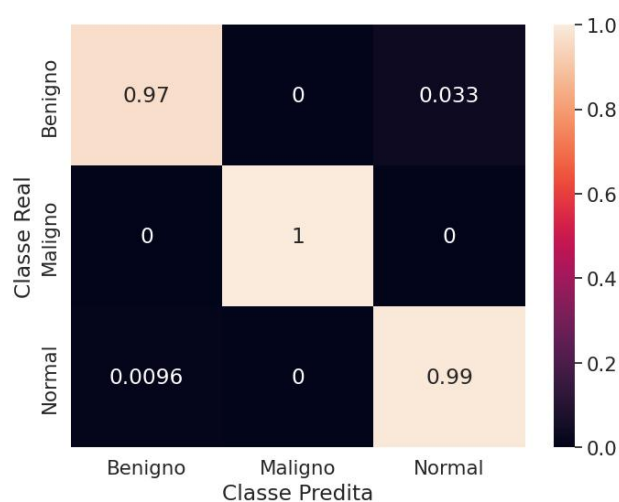


Figura 55 – Dataset Multiclasse - Matriz de Confusão Xception.

Fonte: O Autor

A partir do índice de acertos dos modelos para cada uma das três classes, nota-se que os resultados de ambas as redes, Resnet e Xception, foram satisfatórios, mantendo a alta acurácia e precisão como foi com o conjunto de dados binário. É importante salientar que os erros de classificação que ocorreram foram entre as classes benigno e normal, ou seja, alguns pulmões sem tumor o modelo classificava como contendo tumor benigno, e alguns casos em que o pulmão possuía tumor benigno, o modelo classificou como sendo um pulmão normal (sem tumores). A classe mais sensível, que são os tumores malignos, não foram classificadas de maneira errada por nenhum dos modelos.

5.2.5 Avaliação dos Resultados

Para o modelo Resnet, a acurácia alcançada foi de 0,9927, com uma perda de 0,0920. A precisão, o *f1 Score* e o *recall* também foram altos, indicando um bom desempenho do modelo. Já para o modelo Xception, os resultados obtidos foram semelhantes aos do modelo Resnet, com uma acurácia de 0,9927 e uma perda de 0,0688. Novamente, a precisão, o *f1 Score* e o *recall* foram altos. Os excelentes resultados puderam ser validados visualizando as matrizes de confusão geradas.

Ao analisar os mapas de calor gerados com a técnica de Grad-CAM, podemos notar que assim como foi com o conjunto de dados binário, os modelos conseguiram focar em uma ampla região dos pulmões para realizar as classificações, neste quesito, a rede Xception novamente obteve destaque.

Em resumo, os modelos Resnet e Xception mostraram um desempenho excelente no conjunto de dados multiclasse IQ-OTH/NCCD, com altas taxas de acurácia e precisão. Os resultados destacam a capacidade dos modelos em classificar corretamente os pulmões com tumores malignos, enquanto alguns erros foram observados na distinção entre tumores benignos e pulmões normais. A rede Xception obteve uma leve vantagem em relação a Resnet.

6 Considerações Finais

Na seção de introdução do trabalho, pudemos entender que uma das causas do alto índice de mortalidade de pacientes acometidos de câncer de pulmão é o diagnóstico tardio da doença, sendo assim, estudar formas de se realizar um rápido diagnóstico pode colaborar para a diminuição dessa taxa de mortalidade.

Avançando para a seção de fundamentação teórica, pudemos notar que a utilização de redes neurais convolucionais podem ser bastante adequadas a resolverem problemas de classificação de imagens, sendo assim, podemos aplicá-las na classificação de tumores para auxiliar os profissionais da área da saúde a definí-los como sendo malignos ou benignos.

Já na seção de revisão bibliográfica, pudemos notar que estudos voltados a outros tipos de redes neurais já foram realizados e os resultados foram promissores, o que nos motivou a realizar uma análise de um tipo específico de rede atuando nesse processo, as CNNs.

Na seção de resultados, ao aplicar os modelos baseados nas arquiteturas VGG116, VGG19, Resnet e Xception a um conjunto de dados binário, contendo imagens de tumores malignos ou benignos, pudemos validar que de fato a taxa de assertividade desses modelos são altas, comprovando que as redes neurais convolucionais podem sim ser aplicadas ao problema de classificação de tumores, atuando no processo de descobrimento do câncer de pulmão.

Por conseguinte, também foi possível notarmos que dentre as CNNs, as duas que obtiveram melhores resultados foram as redes Xception e Resnet, respectivamente, o que pôde ser explicado pela quantidade de parâmetros que elas possuem e pelo fato de conseguirem analisar uma área maior das imagens no momento da classificação, como observado através do uso da técnica de Grad-CAM.

Ao aplicarmos essas duas redes, Xception e Resnet, para um problema com multiclases, pudemos verificar que as altas acurácias se mantiveram. Apesar de termos que fazer algumas adaptações no modelo, dado o novo tipo de problema a ser resolvido, essas redes também conseguem ser eficientes para bases de dados com muitas classes. É importante ressaltarmos que novamente a rede Xception obteve o melhor resultado na análise, obtendo uma leve vantagem sobre a rede Resnet.

Além disso, é importante considerar a realização de testes adicionais em conjuntos de dados mais amplos e diversificados, incluindo diferentes tipos de câncer de pulmão e variações nas condições de aquisição das imagens, como diferentes dispositivos de imagem ou protocolos de exame. Isso ajudará a avaliar a robustez e a generalização dos modelos desenvolvidos.

Por fim, é fundamental considerar a implementação e validação clínica dos modelos pro-

postos. Isso envolve a colaboração com profissionais de saúde e a coleta de dados clínicos reais para avaliar o desempenho dos modelos em condições do mundo real. Essa etapa é crucial para garantir a eficácia e a segurança dos modelos antes de sua adoção clínica generalizada.

Em resumo, como trabalhos futuros, podemos considerar:

- Implementar e validar clinicamente os modelos selecionados, como as redes Resnet e Xception, em colaboração com profissionais de saúde;
- Realizar testes adicionais em conjuntos de dados mais amplos e diversificados, considerando diferentes tipos de câncer de pulmão e variações nas condições de aquisição das imagens;
- Realizar validação clínica dos modelos propostos, coletando dados clínicos reais e avaliando o desempenho em condições do mundo real;
- Realizar um estudo para comparar a taxa de acerto das redes neurais com a dos médicos no processo de diagnóstico do câncer de pulmão.

Essas sugestões podem contribuir para avançar o conhecimento na área e fornecer ferramentas mais eficazes para auxiliar no diagnóstico precoce e no tratamento do câncer de pulmão.

Referências

- ALYASRIY HAMDALLA; AL-HUSEINY, M. *The IQ-OTH/NCCD lung cancer dataset*. 2020. Website. Disponível em: <<https://www.kaggle.com/datasets/hamdallak/the-iqothnccd-lung-cancer-dataset?sort=votes>>. Acesso em: 20.12.2022. Citado na página 60.
- CAMPBELL, M.; JR, A. J. H.; HSU, F.-h. Deep blue. *Artificial intelligence*, Elsevier, v. 134, n. 1-2, p. 57–83, 2002. Citado na página 39.
- CARVALHO, A. C. P. d. L. et al. Inteligência artificial: riscos, benefícios e uso responsável. *Estudos Avançados*, SciELO Brasil, v. 35, p. 21–36, 2021. Citado na página 33.
- CDC. *Cancer Statistics At a Glance*. 2021. Website. Disponível em: <<https://gis.cdc.gov/Cancer/USCS/AtAGlance/>>. Acesso em: 06.6.2022. Citado na página 26.
- CHATURVEDI, P. et al. Prediction and classification of lung cancer using machine learning techniques. *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, v. 1099, n. 1, p. 012059, mar 2021. Disponível em: <<https://doi.org/10.1088/1757-899x/1099/1/012059>>. Citado na página 28.
- CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 1251–1258. Citado 3 vezes nas páginas 29, 45 e 47.
- DOU, Q. et al. Multilevel contextual 3-d cnns for false positive reduction in pulmonary nodule detection. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 64, n. 7, p. 1558–1567, 2016. Citado na página 56.
- DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. *ArXiv*, abs/1603.07285, 2016. Citado 4 vezes nas páginas 40, 41, 42 e 44.
- FAWCETT, T. An introduction to roc analysis. *Pattern recognition letters*, Elsevier, v. 27, n. 8, p. 861–874, 2006. Citado na página 79.
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. Upper Saddle River, N.J.: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. Disponível em: <<http://www.amazon.com/Digital-Image-Processing-3rd-Edition/dp/013168728X>>. Citado 4 vezes nas páginas 47, 48, 49 e 50.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado 3 vezes nas páginas 39, 40 e 41.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. [S.l.]: MIT press, 2016. Citado na página 78.
- GURNEY, K. *An Introduction to Neural Networks*. London: CRC press, 1997. 234 p. ISBN 9781315273570. Citado 3 vezes nas páginas 35, 36 e 37.
- HANSON, R.; STUTZ, J.; CHEESEMAN, P. *Bayesian classification theory*. [S.l.], 1991. Citado na página 31.

- HAYKIN, S. *Neural Networks and Learning Machines*. Pearson Education, 2011. ISBN 9780133002553. Disponível em: <<https://books.google.com.br/books?id=faouAAAAQBAJ>>. Citado 4 vezes nas páginas 36, 37, 38 e 39.
- HE, K. et al. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778. Citado 4 vezes nas páginas 28, 44, 45 e 46.
- HUANG, X. et al. Deep transfer convolutional neural network and extreme learning machine for lung nodule diagnosis on ct images. *Knowledge-Based Systems*, Elsevier, v. 204, p. 106230, 2020. Citado 2 vezes nas páginas 55 e 56.
- IBGE. *Ministério do Planejamento, Orçamento e Gestão. Instituto Brasileiro de Geografia e Estatística - Rio de Janeiro. Pesquisa Nacional por Amostra de Domicílios*. 2008. Website. Disponível em: <<https://www.ibge.gov.br/>>. Acesso em: 07.6.2022. Citado na página 27.
- INCA. *O que é câncer?* 2020. Website. Disponível em: <<https://www.inca.gov.br/o-que-e-cancer>>. Acesso em: 15.5.2022. Citado 3 vezes nas páginas 25, 26 e 27.
- INCA. *Como surge o câncer?* 2021. Website. Disponível em: <<https://www.inca.gov.br/como-surge-o-cancer>>. Acesso em: 08.6.2022. Citado na página 25.
- LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group UK London, v. 521, n. 7553, p. 436–444, 2015. Citado na página 78.
- LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: IEEE. *Proceedings of 2010 IEEE international symposium on circuits and systems*. [S.l.], 2010. p. 253–256. Citado na página 41.
- MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizado de máquina. *Sistemas inteligentes-Fundamentos e aplicações*, Manole, v. 1, n. 1, p. 32, 2003. Citado 5 vezes nas páginas 31, 32, 33, 51 e 52.
- NASSER, I. M.; ABU-NASER, S. S. Lung cancer detection using artificial neural network. *International Journal of Engineering and Information Systems (IJEAIS)*, v. 3, n. 3, p. 17–23, 2019. Citado na página 28.
- NISHIO, M. et al. Computer-aided diagnosis of lung nodule using gradient tree boosting and bayesian optimization. *PLOS ONE*, Public Library of Science, v. 13, n. 4, p. 1–13, 04 2018. Disponível em: <<https://doi.org/10.1371/journal.pone.0195875>>. Citado 2 vezes nas páginas 55 e 56.
- OMS. *Cancer Survival*. 2021. Website. Disponível em: <<https://gco.iarc.fr/>>. Acesso em: 15.5.2022. Citado 2 vezes nas páginas 26 e 27.
- Poreva et. al, A. P. Y. K. V. V. Machine learning techniques application for lung diseases diagnosis. In: IEEE. *2017 5th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. [S.l.], 2017. p. 1–5. Citado 2 vezes nas páginas 55 e 56.
- QUEIROZ, J. E. R. de; GOMES, H. M. Introdução ao processamento digital de imagens. *Rita*, v. 13, n. 2, p. 11–42, 2006. Citado 2 vezes nas páginas 49 e 50.

- RESEARCH, C. *Health professional - Data and Statistics - Cancer Statistics for the UK*. 2018. Website. Disponível em: <<https://www.cancerresearchuk.org/health-professional/cancer-statistics-for-the-uk>>. Acesso em: 08.6.2022. Citado na página 28.
- RODRIGUES, M. B. et al. Health of things algorithms for malignancy level classification of lung nodules. *IEEE Access*, IEEE, v. 6, p. 18592–18601, 2018. Citado 3 vezes nas páginas 28, 55 e 56.
- ROSSUM, G. V.; DRAKE, F. L. *An introduction to Python*. [S.l.]: Network Theory Ltd. Bristol, 2003. Citado na página 61.
- SELVARAJU, R. R. et al. Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2017. p. 618–626. Citado na página 53.
- SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. Citado 4 vezes nas páginas 28, 43, 44 e 45.
- SONG, Q. et al. Using deep learning for classification of lung nodules on computed tomography images. *Journal of healthcare engineering*, Hindawi, v. 2017, 2017. Citado na página 41.
- TEODORO, A.; GOMES, O. Desenvolvimento de rede neural artificial para distribuição de chaves em algoritmos de criptografia simétrica utilizando hardware reconfigurável. In: . [S.l.: s.n.], 2018. Citado na página 35.
- TRAN, G. S. et al. Improving accuracy of lung nodule classification using deep learning with focal loss. *Journal of healthcare engineering*, Hindawi, v. 2019, 2019. Citado na página 56.
- WEISS, S. M.; KULIKOWSKI, C. A. *Computer systems that learn: classification and prediction methods from statistics, neural nets, machine learning, and expert systems*. [S.l.]: Morgan Kaufmann Publishers Inc., 1991. Citado na página 52.