UNIVERSIDADE FEDERAL FLUMINENSE

BRUNO SOARES DE BARROS

Combinando Redes Neurais Convolucionais e LSTM para a Classificação de COVID-19 e Pneumonia Bacteriana em Vídeos de Ultrassom Pulmonar

NITERÓI 2022

BRUNO SOARES DE BARROS

Combinando Redes Neurais Convolucionais e LSTM para a Classificação de COVID-19 e Pneumonia Bacteriana em Vídeos de Ultrassom Pulmonar

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

Orientadora: Aura Conci

Coorientador: Célio Vinicius Neves de Albuquerque

> NITERÓI 2022

Ficha catalográfica automática - SDC/BEE Gerada com informações fornecidas pelo autor

B277c Barros, Bruno Soares de Combinando Redes Neurais Convolucionais e LSTM para a Classificação de COVID-19 e Pneumonia Bacteriana em Vídeos de Ultrassom Pulmonar / Bruno Soares de Barros ; Aura Conci, orientadora ; Célio Vinicius Neves de Albuquerque, coorientador. Niterói, 2022. 163 f.
Dissertação (mestrado)-Universidade Federal Fluminense, Niterói, 2022.
DOI: http://dx.doi.org/10.22409/PGC.2022.m.08698745784
1. Aprendizado profundo. 2. Visão computacional. 3. Rede neural convolucional. 4. Rede neural recorrente. 5. Produção intelectual. I. Conci, Aura, orientadora. II. Albuquerque, Célio Vinicius Neves de, coorientador. III. Universidade Federal Fluminense. Instituto de Computação. IV. Título.

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

BRUNO SOARES DE BARROS

Combinando Redes Neurais Convolucionais e LSTM para a Classificação de COVID-19 e Pneumonia Bacteriana em Vídeos de Ultrassom Pulmonar

> Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Mestre em Computação. Área de concentração: Ciência da Computação

> > _____

Aprovada em janeiro de 2022.

BANCA EXAMINADORA

Profa. Dra. Aura Conci - orientadora, UFF

Prof. Dr. Célio V. N. de Albuquerque - coorientador, UFF

Rofa. Dra. Afine Marine Paes Carvalho, UFF

Deerhurtu

Profa. Dra. Débora C. Muchaluat Saade, UFF

ANSELMO CARDOSO DE PAIVA:37552384387 Dados: 2022.01.21 06:59:32-03:00

> Prof. Dr. Anselmo Cardoso de Paiva, UFMA ARISTOFANES CORREA SILVA:28874536372 Assinado de forma digital por ARISTOFANES CORREA SILVA:28874536372 Dados: 2022.01.21 08:49:56-03:00'

Prof. Dr. Aristófanes Corrêa Silva, UFMA

Niterói 2022

Dedicado às vítimas da COVID-19 e aos profissionais da área da saúde.

Agradecimentos

Em primeiro lugar, eu gostaria de agradecer à Universidade Federal Fluminense e ao Instituto de Computação por terem me acolhido e proporcionado um ambiente propício para o desenvolvimento das minhas atividades acadêmicas.

Aos meus orientadores Aura Conci e Célio Albuquerque por acreditarem no meu potencial.

Aos professores e colaboradores da universidade, que mesmo nos momentos mais difíceis — como no pico da pandemia de COVID-19 — mantiveram bravamente o padrão de qualidade e excelência.

Aos colegas de curso Paulo Lacerda e Augusto Araújo por terem me apoiado até o último momento, muitas vezes ajudando a revisar e a melhorar esta dissertação.

Aos membros da banca por todas as sugestões e considerações.

Resumo

Aprendizado profundo é uma técnica importante para a construção de aplicativos de diagnóstico auxiliado por computador. Este trabalho apresenta um método computacional para classificar vídeos de ultrassom pulmonar capturados por transdutores convexos para auxiliar no diagnóstico de COVID-19 e pneumonia bacteriana. Redes neurais convolucionais foram utilizadas para extração das características espaciais e a dependência temporal foi aprendida por uma rede neural recorrente do tipo long short-term memory. Diferentes tipos de arquiteturas de redes neurais convolucionais foram testadas para extração de características. Os hiperparâmetros foram otimizados utilizando o framework Optuna. A extração de características utilizando transferência de aprendizado com redes pré-treinadas no conjunto de dados ImageNet foi semelhante às redes pré-treinadas em imagens de ultrassom pulmonar de outros trabalhos. Os resultados foram comparados com outros estudos, mostrando que o aprendizado profundo e o ultrassom pulmonar podem auxiliar no diagnóstico de COVID-19 e de outras doenças pulmonares. O melhor classificador apresentou uma acurácia média de 95%, precisão de 94,44%, sensibilidade de 100%, especificidade de 95,65%, e F1-Score de 96,77% para o diagnóstico de COVID-19 em uma validação cruzada com cinco partições, superando classificadores baseados em abordagens puramente espaciais.

Palavras-chave: Aprendizado Profundo, CNN, COVID-19, LSTM, Otimização de Hiperparâmetros, RNN, Ultrassom Pulmonar.

Abstract

Deep Learning is an important technique for building computer-aided diagnosis applications. This work presents a computational method to classify lung ultrasound videos captured by convex transducers to aid in the diagnosis of COVID-19 and bacterial pneumonia. Convolutional Neural Networks were used to extract spatial features, and the temporal dependence was learned using a recurrent neural network of type long shortterm memory. Different types of convolutional neural network architectures were tested for feature extraction. The hyperparameters were optimized using the Optuna framework. Feature extraction using transfer learning with networks pre-trained on the ImageNet dataset was similar to networks pre-trained on lung ultrasound images from other works. The outcomes were compared with other studies, showing that deep learning and lung ultrasound can aid in the diagnosis of COVID-19 and other lung diseases. The best classifier presented an average accuracy of 95%, precision of 94.44%, sensitivity of 100%, specificity of 95.65%, and F1-Score of 96.77% for the diagnosis of COVID-19 in a 5-fold cross validation, outperforming classifiers based on purely spatial approaches.

Keywords: CNN, COVID-19, Deep Learning, Hyperparameter Optimization, LSTM, Lung Ultrasound, RNN.

Lista de Figuras

1	Transdutores para USP. Adaptado de (BAKHRU; SCHWEICKERT, 2013).	20
2	Modos de exibição das imagens de ultrassom. Adaptado de (DINH, 2021).	21
3	Achados relacionados à COVID-19, pneumonia bacteriana e pulmão saudável.	22
4	Posições anatômicas. Adaptado de (OLIVEIRA, 2020)	24
5	Regiões importantes no USP, decúbito dorsal (A) e lateral (B). Adaptado de (OLIVEIRA, R. R. de et al., 2020)	24
6	Representação da árvore de decisão do protocolo BLUE	25
7	Representação da interconexão entre neurônios e seus componentes. Adap- tado de (WIKIPEDIA, 2021)	27
8	Representação do modelo de um neurônio artificial (perceptron)	28
9	Exemplo da arquitetura de uma rede MLP	30
10	Exemplo de uma operação de convolução. Adaptado de (ZHANG, A. et al., 2021).	33
11	Exemplo do uso do <i>padding</i> . Adaptado de (ZHANG, A. et al., 2021)	34
12	Exemplo do uso do <i>padding</i> e do <i>stride</i> . Adaptado de (ZHANG, A. et al., 2021).	35
13	Exemplo de uma operação de <i>pooling</i> . Adaptado de (ZHANG, A. et al., 2021).	37
14	Arquitetura da LeNet-5. Adaptado de (LECUN et al., 1998)	38
15	Representação de uma RNN desenrolada. Adaptado de (GOODFELLOW; BENGIO; COURVILLE, 2016)	40
16	Representação de unidades RNN e LSTM. Adaptado de (DONAHUE et al., 2017).	41

17	Exemplo da extração de características utilizando uma VGG16	45
18	Validação cruzada pelo método K-fold com k partições	47
19	Estrutura da matriz de confusão.	51
20	Etapas do processo de extração de conhecimento dos vídeos	80
21	Distribuição dos vídeos em relação ao sexo e diagnóstico do paciente. $\ .\ .$	83
22	Distribuição dos vídeos considerando a idade e diagnóstico do paciente. $\ .$.	84
23	Distribuição dos vídeos em relação a sintomas e achados	85
24	Exemplo de informações removidas	86
25	Os 5 frames extraídos usando a configuração 1	87
26	As primeiras camadas convolucionais da arquitetura Xception	90
27	Extração e classificação combinando uma VGG e LSTM	93
28	Acurácia.	100
29	Precisão, sensibilidade, especificidade e F1-Score para a classe COVID-19 $\stackrel{\circ}{\cdot}$	101
30	Precisão, sensibilidade, especificidade e F1- <i>Score</i> para a classe pneumonia bacteriana	101
31	Precisão, sensibilidade, especificidade e F1-Score para a classe saudável :	102

Lista de Tabelas

1	Avaliação qualitativa do coeficiente de correlação
2	Quantidade de vídeos de ultrassom incluídos neste estudo 83
3	Estatísticas sobre a quantidade de <i>frames</i> do conjunto de dados
4	Quantidade de <i>frames</i> extraídos por configuração
5	Dimensões da camada de entrada para cada uma das CNNs utilizadas 88
6	Dimensão do vetor de características das CNNs
7	Distribuição dos vídeos nas partições de treinamento e teste
8	Camada de entrada da LSTM
9	Valores utilizados para a otimização dos hiperparâmetros 95
10	Os 10 melhores classificadores
11	Resultados do teste de correlação <i>tau</i> de Kendall
12	Resultados da Xception-LSTM e POCOVIDNet-4-LSTM
13	Comparação entre Xception-LSTM e POCOVIDNet-4-LSTM 104
14	Resultados da POCOVIDNet-1-LSTM
15	Resultados da Xception-LSTM e NASNetMobile-LSTM
16	Resultados da Xception-LSTM, POCOVIDNet-4-LSTM, POCOVID-Net e Models Genesis
17	Comparação com os resultados preliminares da Xception-LSTM 107
18	Resumo dos trabalhos relacionados
19	Hiperparâmetros da configuração 1 (5 <i>frames</i>)
20	Hiperparâmetros da configuração 2 (10 <i>frames</i>)
21	Hiperparâmetros da configuração 3 (15 <i>frames</i>)

22	Hiperparâmetros da configuração 4 (20 <i>frames</i>)
23	Parâmetros e classificadores da configuração 1 (5 <i>frames</i>)143
24	Parâmetros e classificadores da configuração 2 (10 frames)
25	Parâmetros e classificadores da configuração 3 (15 frames)
26	Parâmetros e classificadores da configuração 4 (20 <i>frames</i>)
27	Resultados da configuração 1 (5 <i>frames</i>)
28	Resultados da configuração 2 (10 <i>frames</i>)
29	Resultados da configuração 3 (15 <i>frames</i>)
30	Resultados da configuração 4 (20 <i>frames</i>)

Lista de Abreviaturas e Siglas

Adam Adaptive Moment Estimation

- **ANN** Artificial Neural Network
- AUC Area Under ROC Curve

BLUE Bedside Lung Ultrasound in Emergency

CADx Computer-Aided Diagnosis

CNN Convolutional Neural Network

 ${\bf CTC}\,$ Camada Totalmente Conectada

DICOM Digital Imaging and Communications in Medicine

EPH Edema Pulmonar Hidrostático

Espec. Especificidade

FCNN Fully Connected Neural Network

FN Falso Negativo

 ${\bf FP}\,$ Falso Positivo

FPS Frames Per Second

GAP Global Average Pooling

GPU Graphic Processing Unit

GRU Gated Recurrent Units

HDF5 Hierarchical Data Format 5

HPO Hyperparameter Optimization

- ICLUS-DB Italian COVID-19 Lung US Database
- **LSTM** Long-Short Term Memory
- MLP Multilayer Perceptron
- MPEG-4 Moving Picture Experts Group 4
- **MSE** Mean Square Error
- **N-CLAHE** Contrast Limited Adaptive Histogram Equalization
- **OMS** Organização Mundial da Saúde
- **PNG** Portable Network Graphics
- Prec. Precisão
- **R-CNN** Region Based Convolutional Neural Networks
- **Reg-STN** Regularised Spatial Transformer Networks
- **ReLU** Rectified Linear Function
- **RNN** Recurrent Neural Network
- RT-PCR Real-time Polymerase Chain Reaction
- ${\bf RX}\,$ Raio-X
- SDRA Síndrome do Desconforto Respiratório Agudo
- Sens. Sensibilidade
- SGD Stochastic Gradient Descent
- SORD Soft Ordinal Regression
- **SSD** Single Shot Detector
- **STN** Spatial Transformer Network
- **TA** Taxa de Aprendizado
- tanh Hyperbolic Tangent

- ${\bf TC}\,$ Tomografia Computadorizada
- ${\bf TL}\,$ Tamanho do Lote
- **TPE** Tree-Structured Parzen Estimator

 ${\bf TVN}\,$ Taxa de Verdadeiros Negativos

- **TVP** Taxa de Verdadeiros Positivos
- $\mathbf{US} \ \mathrm{Ultrassom}$
- ${\bf USP}~$ Ultrassom Pulmonar
- ${\bf VC}~{\rm Visão}~{\rm Computational}$
- $\mathbf{VGG}\ \ \textit{Visual Geometry Group}$
- ${\bf VN}\,$ Verdadeiro Negativo
- ${\bf VP}~$ Verdadeiro Positivo
- ${\bf VPP}\,$ Valor Preditivo Positivo

Sumário

1	Intr	odução		12
	1.1	Objet	ivos	14
		1.1.1	Objetivos Específicos	14
	1.2	Estrut	tura da Dissertação	15
2	Fun	dament	tação Teórica	16
	2.1	Ultras	som	16
		2.1.1	Sistema de Imagem	17
		2.1.2	Transdutores	18
			2.1.2.1 Profundidade	18
			2.1.2.2 Resolução do Sistema de Imagem	18
			2.1.2.3 Tipos de Transdutores	19
		2.1.3	Modos de Exibição das Imagens	20
		2.1.4	Vídeos de Ultrassom	20
		2.1.5	Ultrassom Pulmonar	21
			2.1.5.1 Achados de Ultrassom Pulmonar	22
			2.1.5.2 Exame	23
			2.1.5.3 Protocolo BLUE	23
	2.2	Redes	Neurais Artificiais	25
		2.2.1	Neurônio	26
		2.2.2	Rede Perceptron	26
		2.2.3	Rede Perceptron Multicamada	29

	2.2.4	Backpropagation
2.3	Redes	Neurais Convolucionais
	2.3.1	Operação de Convolução
	2.3.2	Padding
	2.3.3	<i>Stride</i>
	2.3.4	Camada Convolucional
	2.3.5	Camada de <i>Pooling</i>
	2.3.6	Camada de <i>Global Average Pooling</i>
	2.3.7	Camada de <i>Flatten</i>
	2.3.8	Arquiteturas de Redes Neurais Convolucionais
2.4	Redes	Neurais Recorrentes
	2.4.1	Long Short-Term Memory
		2.4.1.1 Estado da Célula
		2.4.1.2 Forget Gate
		2.4.1.3 Input Gate e Input Modulation Gate
		2.4.1.4 <i>Output Gate</i>
2.5	Transf	erência de Aprendizado
	2.5.1	Métodos de Transferência de Aprendizado 44
		2.5.1.1 Método sem Ajuste Fino
		2.5.1.2 Método com Ajuste Fino
		2.5.1.3 Extração de Características
2.6	Sobrea	juste
	2.6.1	Camada de <i>Dropout</i>
2.7	Valida	ção Cruzada
2.8	Funçõ	s de Ativação
	2.8.1	Sigmoide

		2.8.2	Softmax	48
		2.8.3	Tanh	48
		2.8.4	ReLU	48
	2.9	Função	o de Perda	49
	2.10	Otimiz	ação de Hiperparâmetros	49
	2.11	Avalia	ção dos Classificadores	50
		2.11.1	Acurácia	51
		2.11.2	Precisão	51
		2.11.3	Sensibilidade	51
		2.11.4	Especificidade	52
		2.11.5	F1-Score	52
		2.11.6	Coeficiente de Correlação	52
			2.11.6.1 Coeficiente de Correlação de Kendall	53
			2.11.6.2 Construção do Teste de Hipóteses	54
3	Trab	oalhos F	2.11.6.2 Construção do Teste de Hipóteses	54 55
3	Trab 3.1	balhos F Doenç	2.11.6.2 Construção do Teste de Hipóteses	54 55 55
3	Trab 3.1	balhos F Doenç 3.1.1	2.11.6.2 Construção do Teste de Hipóteses	545555
3	Trab 3.1	Dalhos F Doenç 3.1.1 3.1.2	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57
3	Trab 3.1	Doenç. 3.1.1 3.1.2 3.1.3	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58
3	Trab 3.1 3.2	Doenç 3.1.1 3.1.2 3.1.3 COVII	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58 60
3	Trab 3.1 3.2	Doenç. 3.1.1 3.1.2 3.1.3 COVII 3.2.1	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58 60 60
3	Trab 3.1 3.2	Doenç 3.1.1 3.1.2 3.1.3 COVII 3.2.1	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58 60 60 60 60
3	Trab 3.1 3.2	Doenç. 3.1.1 3.1.2 3.1.3 COVII 3.2.1	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58 60 60 60 62
3	Trab 3.1 3.2	Doenç 3.1.1 3.1.2 3.1.3 COVII 3.2.1	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58 60 60 60 62 64
3	Trab 3.1 3.2	Doenç. 3.1.1 3.1.2 3.1.3 COVII 3.2.1	2.11.6.2 Construção do Teste de Hipóteses	 54 55 55 57 58 60 60 60 62 64 64

			3.2.2.3 Awasthi et al. (2021)	67
			3.2.2.4 Muhammad e Hossain (2021) $\ldots \ldots \ldots \ldots \ldots$	70
		3.2.3	Outros Conjuntos de Dados	72
			3.2.3.1 Jiaqi Zhang et al. (2020)	72
			3.2.3.2 Tsai et al. (2021)	74
			3.2.3.3 Arntfield et al. (2021)	75
	3.3	Conclu	1sões	77
		3.3.1	Classificação em Nível de Frame	77
		3.3.2	Classificação em Nível de Vídeo	78
	3.4	Contri	buições do Trabalho	78
4	Mát	odo nar	a Classificação de Vídeos de Ultrassom Pulmonar	80
-	4 1		a classificação de Videos de Ortrassoni i unifonai	00
	4.1	Consti	rução do Conjunto de Dados	81
		4.1.1	Pesquisa do Conjunto de Dados	81
		4.1.2	Vídeos de Ultrassom Utilizados nesta Dissertação	82
		4.1.3	Caracterização do Conjunto de Dados	83
	4.2	Pré-pr	ocessamento dos Dados	85
		4.2.1	Pré-processamento dos Vídeos	86
		4.2.2	Pré-processamento dos <i>Frames</i>	88
	4.3	Proces	samento dos Dados	89
		4.3.1	Extração das Características Espaciais	89
	4.4	Classif	ficação	90
		4.4.1	Particionamento do Conjunto de Dados	91
		4.4.2	Aprendizado das Características Temporais	92
		4.4.3	Treinamento e Otimização dos Hiperparâmetros dos Classificadores	94
	4.5	Impac	to das Configurações de Extração dos <i>Frames</i>	95

5	Resultados			
	5.1	Treinamento e Otimização dos Hiperparâmetros dos Classificadores	97	
	5.2	Avaliação dos Classificadores	98	
	5.3	Impacto das Configurações de Extração dos Frames	99	
6	Disc	ussões	103	
	6.1	Treinamento e Otimização dos Hiperparâmetros dos Classificadores	103	
	6.2	Avaliação dos Classificadores	103	
	6.3	Impacto das Configurações de Extração dos Frames	105	
	6.4	Comparação com o Trabalho de Referência	106	
	6.5	Outras Comparações	106	
7	Con	clusões	108	
	7.1	Limitações do Trabalho	109	
	7.2	Trabalhos Futuros	110	
RI	EFER	ÊNCIAS	112	
Ap	ôêndi	ce A – Resumo dos Trabalhos Relacionados	132	
Ap	ôndio	ce B – Hiperparâmetros dos Modelos	140	
Ap	ôndio	ce C - Número de Parâmetros e Tamanho dos Classificadores	143	
Ap	Apêndice D – Resultado da Avaliação dos Classificadores 14			

1 Introdução

Desde os primeiros casos do coronavírus (COVID-19), em dezembro de 2019, registrados em um hospital na cidade de Wuhan, China (ZHU, N. et al., 2020; PROMED, 2021), mais de 270 milhões de casos confirmados e 5 milhões de mortes em todo o mundo foram notificados à Organização Mundial da Saúde (OMS) (WHO, 2021). A doença continua a se espalhar e novas variantes do vírus, algumas delas mais resistentes à neutralização de anticorpos apareceram (RESENDE et al., 2021). Essas variantes apresentam maior taxa de transmissão (VOLZ et al., 2021; SABINO et al., 2021; KATELLA, 2021), levando ao surgimento de surtos epidêmicos em várias partes do globo (VAIDYANATHAN, 2021), impactando o sistema de saúde.

Segundo a literatura científica, o vírus SARS-CoV-2 (sigla do inglês que significa coronavírus 2 da síndrome respiratória aguda grave) causador da doença COVID-19, é transmitido principalmente de pessoa para pessoa, por vias aéreas e superfícies contaminadas (KIM et al., 2020; LA ROSA et al., 2020). Esse tipo de transmissão dificulta o trabalho da equipe médica, que precisa estar atenta aos procedimentos e protocolos de atendimento para evitar a contaminação da equipe e das demais pessoas.

Estudos baseados em pacientes afetados pela COVID-19 indicam uma alta prevalência de sintomas respiratórios que requerem uma avaliação mais profunda (CHEN et al., 2020; HUANG, R. et al., 2020). A investigação inicial da doença normalmente envolve o exame clínico, a ausculta pulmonar e exames complementares de imagem, como o raio-X (RX) e a tomografia computadorizada (TC) (BUONSENSO; PATA; CHIARETTI, 2020).

Os exames clínicos e de imagem são propensos à disseminação do vírus, principalmente devido ao número de pessoas da equipe médica mobilizadas para examinar e transportar o paciente para os centros onde estão localizados os equipamentos de diagnóstico por imagem. Outro aspecto que deve ser considerado é a possibilidade de propagação do vírus pela contaminação do equipamento e do próprio estetoscópio, que precisam ser esterilizados a cada exame (BUONSENSO; PATA; CHIARETTI, 2020). Além do risco de contaminação, há o fato de o estetoscópio e a ausculta nesses casos serem de baixa utilidade comprovada (BUONSENSO; PATA; CHIARETTI, 2020). Estudos indicam que o ultrassom pulmonar (USP) pode superar o padrão atual de atendimento, tanto em agilidade quanto no diagnóstico em casos de insuficiência respiratória (sintoma grave da COVID-19) (WALDEN et al., 2018).

O USP tem boa sensibilidade na detecção de patologias pulmonares (AMATYA et al., 2018; GIBBONS et al., 2021). Em relação à COVID-19, estudos relatam uma alta correlação entre os achados radiológicos do USP e do exame de TC de tórax (BRAHIER et al., 2020; OLIVEIRA, R. R. de et al., 2020; DEMI, 2020; TUNG-CHEN et al., 2020; ZHU, F. et al., 2020; PEIXOTO et al., 2021). Nesse sentido, alguns estudos sugerem o uso do ultrassom (US) como alternativa à ausculta e aos exames complementares de RX e TC (YANG et al., 1992; LICHTENSTEIN; GOLDSTEIN et al., 2004; AUJAYEB, 2020; KIAMANESH et al., 2020; MONGODI et al., 2020; NETO; QUEIROZ, 2020).

O US apresenta vantagens que podem ajudar no combate à COVID-19, pois é portátil, livre de radiação, fácil de esterilizar, tem baixo custo de aquisição, permite que o exame seja realizado à beira do leito e pode ser utilizado pelo médico sem a necessidade de mobilizar outros profissionais (MCDERMOTT et al., 2021). Além disso, devido à portabilidade e ao baixo custo de aquisição, tais dispositivos permitem a integração rápida com sistemas que podem fazer uso intensivo de técnicas de inteligência artificial e visão computacional para auxiliar no diagnóstico de COVID-19 (AWASTHI et al., 2021).

As técnicas de aprendizado profundo são atualmente consideradas o estado da arte em muitas aplicações médicas e de visão computacional (VC). Conforme a revisão realizada em Shengfeng Liu et al. (2019), muitas aplicações superam ou igualam os resultados obtidos por especialistas humanos. No entanto, embora o uso de aprendizado profundo tenha avançado na área médica, incluindo o diagnóstico assistido por computador (CADx, *Computer-Aided Diagnosis*) (BHATTACHARYA et al., 2021; ESTEVA et al., 2021; SAR-VAMANGALA; KULKARNI, 2021), a aplicação dessas técnicas em imagens de US pode ser considerada incipiente (HUANG; ZHANG; LI, 2018).

Apesar das vantagens apresentadas no uso de imagens de US para o diagnóstico de patologias pulmonares e da COVID-19, poucos estudos exploraram técnicas de aprendizado profundo em comparação ao número de estudos que investigaram essas técnicas em imagens de RX e TC (LIU, X. et al., 2019; DESAI; PAREEK; LUNGREN, 2020; GOZES et al., 2020; HORRY et al., 2020; SWAPNAREKHA et al., 2020; WU et al., 2020; ZHOU, S. K. et al., 2020; AKRAM et al., 2021; ASLAN et al., 2021; JÚNIOR et al., 2021; LA-

CERDA et al., 2021; NARIN; KAYA; PAMUK, 2021; SYEDA et al., 2021; TAYARANI N., 2021; WANG, S. H. et al., 2021; GAYATHRI et al., 2021; HASSAN et al., 2021). Uma das possíveis causas é a defasagem das bases de dados públicas disponíveis para pesquisas científicas, principalmente no que diz respeito aos vídeos de USP contendo casos comprovados da COVID-19 (NGUYEN et al., 2021). Outra possível causa pode estar relacionada com a qualidade dos estudos de US, visto que é uma modalidade dependente da habilidade do operador, existindo a necessidade de uma padronização dos critérios de treinamento e certificação (SHAW; LOUW; KOEGELENBERG, 2020).

Na revisão realizada em Islam et al. (2021), onde especialistas humanos interpretaram imagens de USP referentes a 466 participantes, foi apontada uma sensibilidade de 86,4% para o diagnóstico da COVID-19. No entanto, a especificidade foi de 54,6%, ou seja, a capacidade de diagnosticar corretamente os indivíduos que não apresentavam a COVID-19 foi baixa.

O teste laboratorial denominado RT-PCR (do inglês *Real-time Polymerase Chain Reaction*), é considerado o padrão ouro (OLIVEIRA, B. A. et al., 2020), sendo adotado para o diagnóstico da doença. Entretanto, este teste além de demandar tempo (cerca de dois dias), apresenta resultados com uma baixa sensibilidade ($\approx 70\%$), sendo recomendado, em alguns casos, repetir o teste para a confirmação do diagnóstico (WATSON; WHITING; BRUSH, 2020).

Esses dados reforçam a necessidade de novos estudos que considerem o uso de técnicas de aprendizado profundo utilizando imagens de US, trazendo novas evidências científicas sobre o assunto. Além disso, ainda existe margem para melhorarias, principalmente no que diz respeito ao desenvolvimento de aplicativos de diagnóstico auxiliado por computador (CADx, *Computer-Aided Diagnosis*).

1.1 Objetivos

Este trabalho tem como objetivo propor um método computacional baseado em aprendizado profundo, utilizando as características espaço-temporais presentes nos vídeos de USP para auxiliar no diagnóstico de COVID-19 e pneumonia bacteriana.

1.1.1 Objetivos Específicos

Pode-se destacar os seguintes objetivos específicos:

- Investigar e selecionar uma arquitetura de rede neural convolucional (CNN, *Convolutional Neural Network*) pré-treinada para atuar como um extrator de características espaciais.
- Treinar e otimizar os hiperparâmetros de uma rede neural recorrente (RNN, *Recurrent Neural Network*) do tipo LSTM (do inglês *Long-Short Term Memory*) para o aprendizado de características temporais, que atuará como o classificador.
- Investigar o impacto da quantidade de *frames* extraídos dos vídeos no resultado do método proposto.

1.2 Estrutura da Dissertação

Este trabalho está organizado da seguinte maneira:

- O Capítulo 2 apresenta a fundamentação teórica necessária para compreensão dos métodos utilizados neste trabalho;
- O Capítulo 3 apresenta os trabalhos relacionados e como este trabalho se diferencia deles;
- O Capítulo 4 descreve o método para classificação de vídeos de USP e todas as ferramentas utilizadas;
- No Capítulo 5 são apresentados os resultados obtidos pelo método proposto;
- No Capítulo 6 são apresentadas as discussões;
- No Capítulo 7 são apresentadas as conclusões, limitações e os trabalhos futuros;
- No Apêndice A é apresentada uma tabela contendo um resumo dos trabalhos relacionados;
- No Apêndice B são apresentadas quatro tabelas contendo os melhores conjuntos de hiperparâmetros dos classificadores selecionados no processo de otimização;
- No Apêndice C são apresentadas quatro tabelas contendo informações sobre o número de parâmetros e o tamanho dos classificadores; e
- Por fim, o Apêndice D apresenta quatro tabelas contendo os resultados numéricos da avaliação dos classificadores.

2 Fundamentação Teórica

Este capítulo apresenta o referencial teórico necessário para o entendimento das técnicas e conceitos utilizados nesta dissertação. A Seção 2.1 descreve o uso do ultrassom (US) para fins de diagnóstico médico. As Seções 2.2 a 2.9 abordam os tópicos sobre aprendizado profundo. A Seção 2.10 apresenta o tópico de otimização de hiperparâmetros. A Seção 2.11 define as medidas e testes estatísticos considerados na avaliação dos classificadores.

2.1 Ultrassom

Na área de imagens médicas, o ultrassom é um método que usa ondas sonoras de alta frequência para interagir com os tecidos biológicos e gerar imagens das estruturas internas do corpo (BUI; TAIRA, 2010).

Quase todos os sistemas de ultrassom podem produzir imagens em tempo real, o que é uma grande vantagem para imagens médicas (NAJARIAN; SPLINTER, 2012). No entanto, o uso do ultrassom é mais amplo, sendo utilizado para diferentes aplicações, por exemplo, para verificação de fissuras microscópicas em asas de aviões (SUETENS, 2017).

As ondas sonoras podem ser classificadas segundo sua frequência. Dessa forma, ondas abaixo do intervalo de frequência audível (1–20.000 Hz) são consideradas ondas infrassom e as ondas acima desse intervalo são classificadas como ultrassom, normalmente situadas no intervalo de 1–100 MHz (para fins de diagnóstico) (NAJARIAN; SPLINTER, 2012). A frequência indica quantas ondas completas que se deslocam em determinada velocidade ocorrem em uma unidade de tempo, conforme apresentado na Equação 2.1.

$$f = \frac{v}{\lambda} \tag{2.1}$$

Onde f é a frequência (Hz), v a velocidade da onda no meio em que se propaga (m/s), e λ é o comprimento da onda (m). As ondas de ultrassom são ondas mecânicas longitudinais. O movimento do mecanismo que forma a onda é paralelo à direção de propagação e sua energia depende do movimento das partículas no meio, não se propagando no vácuo (NAJARIAN; SPLIN-TER, 2012).

2.1.1 Sistema de Imagem

A imagem de ultrassom é processada em uma escala de cinza pelo sistema de imagem. Essa escala representa a intensidade do sinal de ultrassom gerado pela diferença da impedância acústica devido à propagação da onda pelos tecidos, formando assim a imagem das estruturas internas. Dessa forma, quanto maior a intensidade do sinal recebido pelo transdutor (eco), maior será o brilho da imagem (NAJARIAN; SPLINTER, 2012).

A Equação 2.2 define a impedância acústica de um tecido.

$$Z = \rho \times c \tag{2.2}$$

Onde Z é a impedância acústica (kg/m² s), ρ é a densidade do meio (kg/m³) e c a velocidade de propagação da onda no meio em que se propaga (m/s).

O coeficiente de reflexão (R) é dado pela relação entre a impedância acústica de dois tecidos $(Z_1 \in Z_2)$, conforme a Equação 2.3, onde $Z_1 \in Z_2$ representam a impedância acústica dos tecidos

$$R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \tag{2.3}$$

A interação da onda acústica nas interfaces, provoca além da reflexão da onda, a transmissão dela. Parte da energia é transmitida conforme a Equação 2.4, sendo T o coeficiente de transmissão e R o coeficiente de reflexão. Para $Z_1 = Z_2$ a onda será completamente transmitida e para os casos onde $Z_1 << Z_2$, a onda será quase que toda refletida (BUSHBERG; BOONE, 2011).

$$T = 1 - R \tag{2.4}$$

2.1.2 Transdutores

Os transdutores são compostos, normalmente, por materiais piezoelétricos que possuem a propriedade de converter sinais elétricos em ondas acústicas e vice-versa. Parte dessa onda acústica transmitida pelo transdutor será refletida (eco) nas interfaces devido à diferença de impedância acústica dos tecidos. O sinal refletido é recebido pelo transdutor, que converterá o sinal analógico em digital para ser processado pelo sistema de imagem (NAJARIAN; SPLINTER, 2012). Logo, os transdutores têm um papel fundamental na qualidade das imagens geradas pelo sistema de imagem e, portanto, a escolha de um tipo específico de transdutor é um aspecto importante a ser considerado pelo operador do US.

Outro aspecto importante a ser considerado é a frequência dos transdutores. A frequência influencia na profundidade das estruturas internas que podem ser visualizadas e na resolução do sistema de imagem de ultrassom, conforme será explicado nas próximas seções.

2.1.2.1 Profundidade

A profundidade está relacionada ao quão superficial ou profunda está localizada uma estrutura interna que se pretende visualizar. Quanto mais profunda está localizada a estrutura, mais difícil será de detectá-la. À medida que a onda de ultrassom se propaga pelos tecidos, ela sofre atenuação, ou seja, a energia da onda é convertida em calor devido ao atrito gerado pelo deslizamento das células e estruturas, transformando a energia mecânica em energia térmica.

A atenuação em tecidos biológicos é quase proporcional a frequência da onda de ultrassom no intervalo de 1–6 MHz (NAJARIAN; SPLINTER, 2012), no qual é um intervalo normalmente utilizado em imagens médicas de ultrassom. Dessa forma, quanto maior a frequência do transdutor, maior será a atenuação da onda ao se propagar pelos tecidos, dificultando a detecção pelo transdutor do sinal refletido (eco) por estruturas mais profundas.

2.1.2.2 Resolução do Sistema de Imagem

A resolução do sistema de imagem dividi-se em resolução axial (direção do feixe) e resolução lateral (perpendicular ao feixe), estas dependentes das características do transdutor.

A resolução axial é o nível de distinção entre as camadas subsequentes na direção

da propagação da onda, dependendo diretamente do comprimento dela, visto que os sinais refletidos são detectados somente quando a onda está em seu pico máximo (crista) ou mínimo (vale) (NAJARIAN; SPLINTER, 2012). A resolução lateral é determinada pela largura do feixe de ultrassom que se propaga através dos tecidos e, portanto, são dependentes das dimensões físicas do transdutor.

2.1.2.3 Tipos de Transdutores

Existem diferentes tipos de transdutores, porém os mais utilizados na ultrassonografia pulmonar são os apresentados na Figura 1. Eles se baseiam em sensores organizados em linha e permitem a obtenção de imagens 2D em corte transversal. Essa linha é composta por cristais piezoelétricos dispostos ao longo da direção azimutal e pode ser classificada como linear, convexa e *phased* (LEE; ROH, 2017), conforme se observa na Figura 1. Cada um desses tipos tem as seguintes características:

- Transdutor linear: é composto por um arranjo linear de cristais piezoelétricos capazes de gerar um feixe de ultrassom para fazer a varredura em linha. A imagem de ultrassom resultante deste arranjo é uma imagem retangular (Figura 1A). Esse tipo de transdutor é mais utilizado para imagens mais precisas (maior resolução axial) e possuem uma frequência mais elevada (7–18 MHz). É destinado aos exames de estruturas mais externas e superficiais (menor profundidade) (LEE; ROH, 2017);
- Transdutor convexo: os cristais piezoelétricos são dispostos de uma forma curva ao longo da sua superfície externa chamada de direção azimutal. O método de aquisição é o mesmo do linear, mas o imageamento resultante possui um formato de leque (Figura 1B). Possuem uma frequência mais baixa (3–6 MHz), com um comprimento de onda acústica mais longa e, portanto, fornecem maior penetração e visualização das estruturas internas mais profundas; e
- Transdutor phased array: ao contrário dos outros dois tipos de transdutores, os transdutores do tipo phased array possuem a vantagem de poder direcionar o feixe de ultrassom ajustando as fases das ondas enviadas pelos cristais individuais (equivalente eletrônico de inclinar o transdutor mecanicamente) (SUETENS, 2017), ou seja, são menos afetados por obstáculos, por exemplo, as costelas. Assim como o transdutor linear, também é composto por um arranjo linear de cristais piezoelétricos e o imageamento resultante possui um formato de cone circular (Figura 1C). Possuem uma frequência mais baixa (2–5 MHz).



Figura 1: Transdutores para USP. Adaptado de (BAKHRU; SCHWEICKERT, 2013).

2.1.3 Modos de Exibição das Imagens

Os dois modos de exibição de imagens mais utilizados para o processamento dos sinais são: o modo B e o modo M. Ambos estão relacionados com a maneira que os sinais refletidos pelas estruturas internas são recebidos e processados. No modo B (modo brilho ou modo bidimensional) os sinais são exibidos como uma imagem 2D convencional, onde a intensidade da energia é demonstrada com pontos de diferentes intensidades. No modo M (modo movimento) as estruturas são acompanhadas, os sinais refletidos são exibidos continuamente ao longo de um eixo vertical (DEXHEIMER NETO et al., 2012). A Figura 2A apresenta uma imagem referente ao modo de processamento B e a Figura 2B refere-se ao modo de processamento M. A linha amarela em ambas as imagens apresentam o eixo vertical utilizado no modo M.

2.1.4 Vídeos de Ultrassom

A velocidade com que as imagens de US são atualizadas é normalmente expressa em FPS (do inglês *Frames Per Second*), ou seja, a quantidade de *frames* que são atualizados no intervalo de um segundo. Uma sequência desses *frames*, por sua vez, dá origem ao que chamamos de vídeo.



Figura 2: Modos de exibição das imagens de ultrassom. Adaptado de (DINH, 2021).

Uma profundidade desnecessariamente grande pode impactar no tempo de espera entre os pulsos, comprometendo a qualidade das imagens e reduzindo a quantidade de *frames* (MITCHELL et al., 2019). Conforme a Seção 2.1.2.1, a profundidade está relacionada ao quão superficial ou profunda está localizada uma estrutura interna que se pretende visualizar. Dessa forma, quanto maior a profundidade, maior será o tempo entre os pulsos, ou seja, o tempo que leva para a onda se propagar até as estruturas mais profundas e retornar até ser detectada pelo transdutor. Nesse sentido, o uso do transdutor adequado pode ajudar a resolver essa questão.

2.1.5 Ultrassom Pulmonar

O pulmão saudável é aerado, cerca de 99% da energia incidente nas interfaces do pulmão (composto por ar e tecidos moles) é refletida (NAJARIAN; SPLINTER, 2012). Isso se deve à grande diferença de impedância acústica entre os meios, como é o caso do ar e dos tecidos moles. Dessa forma, teremos a reflexão quase que total da onda de ultrassom, sendo impossível detectar as estruturas além da pleura, conforme a Equação 2.2.

No entanto, o USP se beneficia das agressões agudas que reduzem essa aeração dos pulmões, alterando sua superfície, tornando possível identificar padrões que passam a ser previsíveis e diagnosticáveis (como em casos de pneumonia) (DEXHEIMER NETO et al., 2012). Essas agressões são identificadas por perfis (A, B e C) e se relacionam com a

quantidade de fluido nos pulmões. Nesse sentido, C indica maior quantidade de fluido nos pulmões, característica das consolidações, por exemplo.

O USP permite elucidar dúvidas na interpretação de diferentes infiltrados pulmonares (substâncias mais densas que o ar), podendo diferenciar o pulmão normal de consolidações, infiltrados intersticial e alveolar ou derrame pleural com boa acurácia (DEXHEIMER NETO et al., 2012). Para o exame de USP, recomenda-se o uso de um transdutor convexo pequeno (3–7 MHz), pois este se adapta melhor aos espaços intercostais, principalmente para o US à beira do leito.

2.1.5.1 Achados de Ultrassom Pulmonar

A Figura 3 apresenta uma série de imagens (Modo B) contendo alguns achados radiológicos que são frequentemente utilizados para avaliação do USP. Esses achados são relacionados à COVID-19, pneumonia bacteriana e pulmão saudável (classes de diagnóstico consideradas neste trabalho), sendo indicados na figura pelas setas vermelhas.



Figura 3: Achados relacionados à COVID-19, pneumonia bacteriana e pulmão saudável.

Dentre os achados mais comuns para COVID-19 encontram-se: as linhas B (Figura 3A); as linhas B coalescentes, representando dentre outros achados, quadros inflamatórios (Figura 3B); o espessamento da linha pleural com irregularidade da pleura e presença de pequenas consolidações em regiões periféricas ou subpleurais (Figura 3C); e derrame pleural, em menor frequência (Figura 3D). As linhas B coalescentes no USP (Figura 3B) possuem correlação com achados da TC, conhecidos como "opacidades em vidro fosco" (OLIVEIRA, R. R. de et al., 2020).

Ao contrário da COVID-19, a pneumonia bacteriana destaca-se por grandes áreas de consolidação, conforme a Figura 3F. Entretanto, outros achados podem ser identificados com maior frequência, como o derrame pleural (Figura 3E).

O pulmão saudável é caracterizado pela presença de uma linha pleural regular (Figura 3H) e por linhas A ou linhas horizontais equidistantes (Figura 3G).

2.1.5.2 Exame

A realização do exame de USP tem por objetivo avaliar a região do pulmão e da pleura através da aquisição de imagens de ultrassom. Para isso, utilizam-se protocolos que auxiliam na delimitação das áreas de estudo e na maneira como o exame é executado e avaliado.

O exame de USP não possui uma posição pré-estabelecida para sua realização, podendo ser executado com o paciente em decúbito dorsal, lateral, ventral, ortostase e sedestação (Figura 4). A posição do paciente para a realização do exame dependerá das suas condições, ou seja, a posição será compatível com o seu quadro clínico. Porém, normalmente o exame é realizado na posição de decúbito dorsal e lateral, conforme a Figura 5.

A região de decúbito dorsal horizontal (Figura 5A) e decúbito lateral esquerdo (Figura 5B) são divididos da seguinte forma: LPS, linha paraesternal; LAA, linha axilar anterior; e LAP, linha axilar posterior. 1, região anterossuperior; 2, região anteroinferior; 3, região lateral superior; 4, região lateral inferior; 5, região posterossuperior; e 6, região posteroinferior.

2.1.5.3 Protocolo BLUE

Existem diferentes protocolos de US descritos na literatura. Para a síndrome respiratória aguda grave, o mais utilizado é o protocolo BLUE (do inglês *Bedside Lung Ultrasound in Emergency Protocol*) (LICHTENSTEIN; MEZIERE, 2008), sendo recomendado para situações de emergência à beira do leito.

O protocolo BLUE descreve uma série de achados, que ao serem combinados resultam em um algoritmo baseado em uma árvore de decisão (Figura 6). Simplificadamente, primeiro verifica-se a presença de deslizamento pleural (nó raiz), depois verifica-se o perfil (linhas A, B e C) seguindo essa lógica pelos nós filhos até chegar a uma folha da árvore, representando uma doença (por exemplo, pneumonia).



Figura 4: Posições anatômicas. Adaptado de (OLIVEIRA, 2020).



Figura 5: Regiões importantes no USP, decúbito dorsal (A) e lateral (B). Adaptado de (OLIVEIRA, R. R. de et al., 2020).

Através dessa árvore de decisão e da predefinição desses perfis e suas localizações é possível diagnosticar doenças respiratórias com uma acurácia média de 90,5% (LICH-TENSTEIN, 2014). No entanto, a utilização desse protocolo dependente da interpretação de um especialista humano.



Figura 6: Representação da árvore de decisão do protocolo BLUE.

2.2 Redes Neurais Artificiais

As redes neurais artificiais (ANNs, do inglês *Artificial Neural Networks*), estão inseridas no contexto da área de inteligência artificial, mais especificamente do aprendizado de máquina e são modelos computacionais inspirados no funcionamento do sistema nervoso central (SNC), seus aspectos e comportamentos (HAYKIN, 2007).

Os primeiros trabalhos sobre as ANNs surgiram na década de 40, com McCulloch e Pitts (1943), seguido por Hebb (1949). Na década seguinte as ANNs ganharam novas contribuições e, em 1958, Rosenblatt apresentou a rede *perceptron* (ROSENBLATT, 1958), uma espécie de neurônio artificial capaz de aprender de maneira supervisionada, atuando como um classificador linear.

No entanto, devido à capacidade computacional limitada da época e das dificuldades

matemáticas, as pesquisas na área foram sendo interrompidas. Além desses fatores, a publicação do livro de Minsky e Papert (1969) teve um papel decisivo para essa interrupção, limitando as redes *perceptron* a problemas linearmente separáveis.

Na década de 80 o interesse pelas ANNs ressurgiu, principalmente devido ao avanço do poder computacional, do crescente interesse pela área de processamento paralelo e da proposição de novas arquiteturas de ANNs. Muitos artigos foram publicados, como os trabalhos de Hopfield (HOPFIELD, 1982; HOPFIELD, 1984) que ajudaram a responder às questões levantadas no final da década de 50 e 60. Em 1986, Rumelhart e McClelland apresentaram um modelo matemático e computacional para o aprendizado supervisionado dos neurônios artificiais (RUMELHART; HINTON; WILLIAMS, 1986; RUMELHART; MCCLELLAND; PDP RESEARCH GROUP, 1986). Surgia o algoritmo para treinar ANNs que ficou conhecido como retropropagação do erro (*backpropagation*), iniciando uma nova era de pesquisa na área.

2.2.1 Neurônio

Os neurônios são as unidades fundamentais do cérebro e do sistema nervoso. Em média possuímos cerca de 100 bilhões de neurônios (CHERNIAK, 1990). Cada neurônio recebe um sinal de entrada através de um elemento chamado dendrito. O neurônio processa a informação e envia outro sinal ao próximo neurônio através do axônio (um prolongamento do neurônio), o que nos leva a uma rede de neurônios interconectados. A Figura 7 apresenta os elementos do neurônio humano.

As conexões entre um terminal axônico (neurônio pré-sináptico) e um dendrito (neurônio pós-sináptico) são chamadas de sinapse. Estima-se que existam cerca de 100 trilhões de sinapses no cérebro humano (INSTITUTE OF MEDICINE; ACKERMAN, 1992). As sinapses atuam mediando as interações entre os neurônios, possuindo um papel excitatório ou inibitório para os impulsos nervosos. Esses impulsos que transitam pela rede interconectada de neurônios possuem um sentido unidirecional, passando pelos dendritos, corpo celular, axônios e terminais sinápticos até chegar ao próximo neurônio e assim por diante.

2.2.2 Rede Perceptron

Em aprendizado de máquina busca-se uma técnica capaz de discriminar os dados, onde dada uma entrada x seja possível mapear para uma saída y. Se o valor de y é conhecido de



Figura 7: Representação da interconexão entre neurônios e seus componentes. Adaptado de (WIKIPEDIA, 2021).

antemão, trata-se de um aprendizado supervisionado. Nesse sentido, pretende-se conhecer a função preditora \hat{f} que mais se aproxime da função f verdadeira, aquela que melhor explica esse mapeamento $f: x \to y$, dado os valores conhecidos de y.

Em 1958, Rosenblatt propôs a rede *perceptron*, um modelo mais simples, porém fundamental para entendimento do funcionamento das ANNs. A ideia por trás da computação realizada por um neurônio artificial, envolve uma soma ponderada dos sinais de entrada. Essa soma corresponde aos valores dos sinais escalados pelas sinapses, simulando o efeito excitatório ou inibitório apresentado pelo modelo biológico. O modelo esquemático de um neurônio de uma rede *perceptron* pode ser visualizado na Figura 8.

O neurônio artificial atua como uma função matemática (também conhecida como função de ativação) aplicando um limiar ao somatório ponderado das entradas somadas a um viés (*bias*). O *bias* (*b*) é o coeficiente linear (*intercept*) da equação linear. O intuito do *bias* é o de aumentar o grau de liberdade desta função e, portanto, mesmo que as entradas sejam nulas, o neurônio pode apresentar uma saída não nula.

Os sinais de entrada são representados por x, onde $x \in \mathbb{R}^{n_x}$ e n_x é o número de sinais de entrada (dimensão). O fator de escala é conhecido como peso (*weight*) sendo representado por w, onde $w \in \mathbb{R}^{n_x}$, visto que para cada sinal x_i teremos um peso w_i associado. A Equação 2.5 apresenta a expressão de saída de um neurônio.


Figura 8: Representação do modelo de um neurônio artificial (perceptron).

$$z = \sum_{i=1}^{n_x} w_i x_i + b$$
 (2.5)

A função de ativação é representada por *a* na Equação 2.6 e a saída da função de ativação por \hat{y} . O conjunto das possibilidades de saída é representado por *y*, onde $y = \{0,1\}$. Dessa forma, \hat{y} assume 1 se z > 0, caso contrário assume 0.

$$\hat{y} = a(z) = \begin{cases} 1, se \ z > 0\\ 0, cc. \end{cases}$$
(2.6)

O aprendizado na rede *perceptron* é realizado alterando-se os valores dos pesos associados às sinapses, de forma que ao ponderar os valores de entrada (x_i) , o resultado da função de ativação (\hat{y}) se aproxime do valor da saída esperada (y) para cada um dos jexemplos do conjunto de treinamento. A atualização dos pesos é descrita pela equação:

$$w_i(t+1) = w_i(t) + \eta x_i^j (y^j - \hat{y}^j)$$
(2.7)

Onde $w_i(t)$ é o peso na *i*-ésima conexão de entrada no instante de tempo t; η é uma taxa de aprendizado (*learning rate*); x_i^j é o valor do *i*-ésimo atributo do *j*-ésimo exemplo do conjunto de treinamento; y^j é a saída esperada (*label*) associada ao *j*-ésimo exemplo do conjunto de treinamento; e \hat{y}^j a saída da função de ativação ou predição. O termo $(y^j - \hat{y}^j)$ na Equação 2.7 representa o erro de classificação do *j*-ésimo conjunto de treinamento.

A taxa de aprendizado (η) contribui na magnitude do ajuste a ser realizado nos pesos (w_i) . Uma taxa alta significa que será necessário realizar grandes alterações no peso e uma taxa baixa implica o contrário. Portanto, a taxa de aprendizado influencia diretamente no tempo de convergência da rede (FACELI et al., 2011).

2.2.3 Rede Perceptron Multicamada

Um dos problemas da rede *perceptron* é a limitação quanto a capacidade de aprendizado. Como visto na Seção 2.2.2, essas redes se limitam a problemas linearmente separáveis. No entanto, em muitos casos reais, esta condição não pode ser satisfeita. Dessa forma, as ANNs do tipo *perceptron* multicamada (MLP, *Multilayer Perceptron*) surgiram como uma solução para esse tipo de problema.

Para problemas não linearmente separáveis, pode-se modificar a arquitetura da ANN para que seja possível aprender funções mais complexas, quebrando essa restrição de linearidade dos dados. Nesse sentido, a rede MLP inclui uma camada a mais, ou seja, uma camada intermediária (também conhecida como camada oculta) entre a camada de entrada e a camada de saída.

Além da modificação em relação à arquitetura, as MLPs utilizam funções de ativação não lineares, como a função *sigmoide* (apresentada na Seção 2.8.1). As funções aprendidas pelos neurônios nas primeiras camadas vão sendo combinadas à medida que a rede tornase mais profunda (com mais camadas). Dessa forma, evita-se que as funções aprendidas nas camadas subsequentes precisem sempre ser apenas combinações lineares das camadas antecedentes.

Essa modificação na arquitetura da rede acaba por adicionar alguns problemas como o sobreajuste dos dados e alto custo computacional, visto que nas MLPs a arquitetura padrão é composta por várias camadas ocultas, onde todos os neurônios de cada uma dessas camadas estão totalmente conectados aos neurônios da camada subsequente. Por esse motivo, esse tipo de rede também é conhecida como uma rede totalmente conectada (FCNN, *Fully Connected Neural Network*).

Os neurônios recebem os dados na camada de entrada (*input layer*) e propagam para a próxima camada conhecida como camada oculta (*hidden layer*). A soma ponderada da saída de uma ou mais camadas ocultas é propagada para a camada de saída (*output layer*) que apresenta o resultado da classificação. A Equação 2.8 representa essa propagação, considerando como entrada para o próximo neurônio o resultado das funções de ativação da camada anterior que serão ponderadas e ativadas até a classificação.

$$a_j^{(l)} = g^{(l)} \left(\sum_k w_{jk}^{(l)} a_k^{(l-1)} + b_j^{(l)}\right) = g^{(l)} (z_j^{(l)})$$
(2.8)

Onde $a_j^{(l)}$ é a função de ativação referente a
oj-ésimo neurônio da l-ésima camada;
 $g^{(l)}$ é a função de ativação da l-ésima camada;
 $w_{jk}^{(l)}$ é o peso referente a conexão entre o
j-ésimo neurônio da l-ésima camada e o
k-ésimo neurônio da camada anterior;
 $a_k^{(l-1)}$ é a função de ativação referente a
ok-ésimo neurônio da camada anterior;
 $a_k^{(l-1)}$ é a função de ativação referente a
ok-ésimo neurônio da camada anterior;
 $a_k^{(l-1)}$ é a função de ativação referente a
ok-ésimo neurônio da camada anterior a l-ésima camada;
 $b_j^{(l)}$ é o bias referente a l-ésima camada do j-ésimo neurônio;
e $z_j^{(l)}$ representa o somatório ponderado das entradas referentes ao j-ésimo neurônio.

A Figura 9 apresenta a arquitetura de uma rede MLP composta por uma camada de entrada, duas camadas ocultas e uma camada de saída, onde $x^{(i)}$ representa os dados da *i*-ésima entrada do conjunto de dados, $a^{(l)}$ é a função de ativação da *l*-ésima camada e $\hat{y}^{(i)}$ o resultado da classificação da *i*-ésima entrada do conjunto de dados.



Figura 9: Exemplo da arquitetura de uma rede MLP.

2.2.4 Backpropagation

O treinamento de uma rede MLP acontece de forma parecida com a rede *percetron*, onde os pesos são atualizados para produzir uma saída (\hat{y}) mais próxima o possível do valor da saída esperada (y). No entanto, esse processo é mais complexo nas MLPs, onde cada camada depende do resultado da computação realizada nas camadas anteriores (Equação 2.8). Dessa forma, para que os pesos possam ser atualizados, o erro da camada de saída precisa ser propagado para as camadas intermediárias.

O algoritmo backpropagation (RUMELHART; HINTON; WILLIAMS, 1986) resolve esse problema, propondo duas etapas para calcular os gradientes da função de perda. Na primeira etapa foward propagation, a entrada é computada pelos neurônios da primeira camada oculta e o resultado da computação é propagado para os neurônios da camada seguinte e assim sucessivamente até a camada de saída (\hat{y}) , onde o erro é computado aplicando-se uma função de perda $L(y,\hat{y})$. A segunda etapa é conhecida como backward propagation, onde o erro é propagado da camada de saída até a primeira camada oculta, no sentido inverso. O ajuste dos pesos é dado pela Equação 2.9:

$$w_{jk}(t+1) = w_{jk}(t) + \eta x_k \delta_j$$
(2.9)

Onde w_{jk} representa o peso entre o *j*-ésimo neurônio e a *k*-ésima saída da camada anterior ou *k*-ésimo atributo de entrada; η representa a taxa de aprendizado; x_k representa a *k*-ésima entrada recebida pelo *j*-ésimo neurônio e δ_j é o erro associado ao *j*-ésimo neurônio.

O erro dos neurônios referentes à camada de saída é conhecido, no entanto, o erro dos neurônios das camadas intermediárias precisa ser calculado. Dessa forma, o erro de um neurônio referente a uma camada intermediária será dado pelo somatório dos valores dos erros dos neurônios da camada seguinte, ponderados pelos pesos das conexões com ele, conforme Equação 2.10:

$$\delta_j = f'_a \sum w_{lj} \delta_l \tag{2.10}$$

Onde f'_a é a derivada parcial da função de ativação do neurônio; w_{lj} é o peso referente a conexão entre *j*-ésimo neurônio e o *l*-ésimo neurônio da camada seguinte; δ_l o erro referente ao *l*-ésimo neurônio. No entanto, se o *j*-ésimo neurônio pertencer à camada de saída o erro será conhecido, e dessa forma adota-se:

$$f_a'e_j \tag{2.11}$$

Onde e_j representa o erro do j-ésimo neurônio da camada de saída.

A contribuição de cada peso no erro da rede para a classificação de um determinado

exemplo de treinamento é dada pela derivada parcial da função de ativação. A atualização dos pesos é feita com base no gradiente descendente dessa derivada, onde se o valor for positivo, então deve-se reduzir o peso e se for negativo então deve-se aumentar o peso (FACELI et al., 2011). A taxa de aprendizado η tem o mesmo papel desempenhado na rede *perceptron*, influenciando diretamente na convergência, aumentando ou diminuindo a magnitude das atualizações dos pesos da rede.

Esses passos se repetem até que um critério de parada seja atingido, por exemplo, uma determinada quantidade de iterações do algoritmo ou um valor mínimo de erro seja satisfeito.

2.3 Redes Neurais Convolucionais

As Rede Neurais Convolucionais (CNNs, *Convolutional Neural Networks*) são um tipo específico de ANNs que processam dados espaciais e estão intimamente ligadas à área de visão computacional (VC). Assim como as ANNs são bio-inspiradas no SNC, as CNNs são bio-inspiradas nas células do córtex visual (LECUN et al., 1998).

Diferentemente das tabelas estruturadas, as imagens apresentam padrões espaciais que podem ser aprendidos e reutilizados. O processamento de dados espaciais se dá devido a uma alteração na arquitetura das MLPs, onde as operações de soma ponderada das entradas realizadas pelos neurônios, são substituídas pelas operações de convolução (muito comuns em processamento de sinais e imagens), dando origem ao nome camada convolucional (GOODFELLOW; BENGIO; COURVILLE, 2016).

Inicialmente, as CNNs foram propostas para problemas de classificação de imagens de dígitos manuscritos (chamadas LeNets) (LECUN et al., 1998), mas devido ao desempenho frente aos resultados obtidos pelas técnicas clássicas de VC, esse tipo de rede acabou rapidamente tornando-se o estado da arte em muitas das aplicações (KHAN, S. et al., 2018).

2.3.1 Operação de Convolução

A operação de convolução aplicada nas camadas convolucionais é uma operação de filtragem no domínio espacial, o que significa dizer que opera diretamente sobre os valores dos *pixels* da imagem de entrada (CONCI; AZEVEDO; LETA, 2008). Dessa forma podemos representar a operação de convolução como:

$$S[i,j] = \sum_{u=-m}^{m} \sum_{v=-n}^{n} K[u,v]I[i-u,j-v]$$
(2.12)

Onde S é a matriz que representa o mapa de características; $i \in j$ são os índices de S; K é a matriz referente ao filtro; $m \in n$ são os índices de K; $u \in v$ são os índices em duas direções ortogonais; e I representa a matriz da imagem de entrada.

A Figura 10 apresenta um exemplo da operação de convolução, onde uma imagem de entrada de dimensões 3×3 *pixels* é convoluída com um filtro de dimensões 2×2 , resultando em um mapa de características de dimensões 2×2 *pixels*. Nesse exemplo, assume-se que o filtro desliza um *pixel* no sentido da largura e um *pixel* no sentido da altura, onde o incremento se dá respectivamente com o movimento do filtro para direita e para baixo, iniciando o movimento no canto superior esquerdo da imagem. Além disso, considera-se que o filtro está alinhado com a imagem, não ultrapassando os limites de suas dimensões. Considerando as condições mencionadas, as dimensões do mapa de características podem ser calculadas por:

$$(n_h - k_h + 1) \times (n_w - k_w + 1) \tag{2.13}$$

Onde $n_h \times n_w$ representam as dimensões da imagem e $k_h \times k_w$ as dimensões do filtro.



Figura 10: Exemplo de uma operação de convolução. Adaptado de (ZHANG, A. et al., 2021).

2.3.2 Padding

O *padding* é utilizado para estender as bordas das imagens. Seu valor é dado em *pixels*, representando o incremento que será realizado nessas bordas. Aplicando um *padding* de 1 *pixel* significa incrementar as dimensões da imagem em 2 *pixels*.

Com a aplicação de sucessivas camadas de convolução, o tamanho dos mapas de características vão sendo sucessivamente reduzidos e consequentemente as informações

dos *pixels* localizados próximo às bordas vão se perdendo. Uma forma de resolver esse problema é usar o *padding*, estendendo-se com zeros a borda da representação da imagem. Dessa forma, a operação de convolução levará em consideração essa nova informação, fazendo com que as dimensões da representação final também aumentem.

A Figura 11 apresenta o uso do *padding* numa operação de convolução, onde uma imagem de entrada de dimensões 3×3 *pixels* e *padding* de 1 *pixel* é convoluída com um filtro de dimensões 2×2 , resultando em um mapa de características de dimensões 4×4 *pixels*. A Equação 2.14 inclui o *padding* para o cálculo das dimensões dos mapas de características.

$$(n_h - k_h + p_h + 1) \times (n_w - k_w + p_w + 1)$$
(2.14)

Onde $n_h \times n_w$ representa as dimensões da imagem; $k_h \times k_w$ as dimensões do filtro; p_h e p_w o incremento nas dimensões da imagem de entrada (*padding*).



Figura 11: Exemplo do uso do *padding*. Adaptado de (ZHANG, A. et al., 2021).

2.3.3 Stride

O stride (ou passo da filtragem) é o parâmetro responsável pelo deslizamento do filtro aplicado na operação de convolução. Seu valor é dado em *pixels*, indicando que o filtro se deslocará essa quantidade de *pixels* em w (largura) e h (altura). Dessa forma, quanto maior o deslizamento, maior será a redução aplicada à imagem de entrada. Para o caso geral (considerando o *filtro*, o *padding* e o *stride*), as dimensões dos mapas de características podem ser calculadas por:

$$\lfloor (n_h - k_h + p_h + s_h)/s_h \rfloor \times \lfloor (n_w - k_w + p_w + s_w)/s_w \rfloor$$

$$(2.15)$$

Onde $n_h \times n_w$ representa as dimensões da imagem; $k_h \times k_w$ as dimensões do filtro; p_w e p_h o incremento em *pixels* nas dimensões da imagem de entrada (*padding*); s_h e s_w a quantidade em *pixels* do deslizamento do filtro (*stride*);

A Figura 12 apresenta o uso do *stride*, onde uma imagem de entrada de dimensões 3×3 *pixels*, *padding* de 1 *pixel* e *stride* de 2 *pixels* é convoluída com um filtro de dimensões 2×2 , resultando em um mapa de características com as dimensões 2×2 *pixels*.



Figura 12: Exemplo do uso do *padding* e do *stride*. Adaptado de (ZHANG, A. et al., 2021).

2.3.4 Camada Convolucional

As camadas convolucionais são utilizadas para extração de características da imagem e utilizam-se das operações de convolução para filtrá-la, ou seja, realçar suas características espaciais. Quando sucessivas operações de convolução são aplicadas ao longo das camadas convolucionais, é possível extrair características cada vez mais complexas (ZHANG, A. et al., 2021).

Em processamento de imagens, para se estabelecer os valores dos coeficientes dos filtros convolucionais é necessário conhecer o objetivo que se deseja alcançar, por exemplo, a redução de ruídos (CONCI; AZEVEDO; LETA, 2008; GONZALEZ; WOODS, 2000). No entanto, uma das vantagens das CNNs é que os coeficientes desses filtros convolucionais podem ser parametrizados e aprendidos no processo de treinamento da rede, da mesma forma que acontece nas redes MLPs (GOODFELLOW; BENGIO; COURVILLE, 2016).

A quantidade de mapas de características que podem ser extraídos da imagem de entrada é determinada pela quantidade de filtros configurados na camada convolucional. As dimensões desses mapas vão depender das dimensões dos filtros convolucionais, do *padding* e do *stride*. Esses são os parâmetros que devem ser configurados na camada convolucional.

Devido às operações de convolução, as conexões da rede tornam-se esparsas, ou seja, cada saída passa a depender somente de uma pequena quantidade de parâmetros da entrada. Essa operação permite reduzir a quantidade de parâmetros envolvidos no treinamento, o que é outra vantagem sobre as MLPs (GU et al., 2018).

2.3.5 Camada de Pooling

A camada de *pooling* (ou subamostragem) é normalmente adicionada após a camada convolucional. Isso se deve ao fato de que, a camada de *pooling* torna os mapas de características subamostrados, invariantes às pequenas translações locais na imagem de entrada (GOODFELLOW; BENGIO; COURVILLE, 2016).

O pooling é um operador assim como a convolução, porém não opera diretamente sobre o valor de um pixel e sim numa vizinhança de pixels considerada. Esse operador atua reduzindo a resolução dos mapas de características, resumindo-os em patches. Na camada de pooling não há parâmetros a serem aprendidos, diferentemente do que acontece com a camada convolucional. No entanto, a operação de pooling também usa um filtro que desliza pelo mapa de características. Dessa forma, uma representação de baixa resolução é criada, mantendo-se os elementos estruturais significativos, mas descartando os detalhes finos.

Os tipos mais utilizados de *pooling* são: o *average pooling* e o *max pooling*. No primeiro caso, o resultado é a média dos valores dos *pixels* sobrepostos pelo filtro e no segundo caso o valor do *pixel* máximo. O resultado será uma imagem de baixa resolução e uma CNN menos suscetível aos efeitos afim (como translação) (SCHERER; MÜLLER; BEHNKE, 2010; BERA; SHRIVASTAVA, 2020).

A Figura 13 apresenta a operação de *pooling*, onde uma imagem de entrada (ou mapa de características) de dimensões 3×3 é subamostrada com um filtro de dimensões 2×2 , resultando em um mapa de características de dimensões 2×2 *pixels* (foi utilizado um *stride* de 1 *pixel*).

2.3.6 Camada de Global Average Pooling

A camada de GAP (do inglês *Global Average Pooling*) é utilizada como uma operação de *average pooling*, conforme apresentado na Seção 2.3.5. No entanto, o papel da camada de GAP é um pouco diferente, pois resume os mapas de características aplicando a média aos valores dos *pixels* referentes a cada um dos mapas de características, resultando em um



Figura 13: Exemplo de uma operação de *pooling*. Adaptado de (ZHANG, A. et al., 2021).

vetor de características. Dessa forma, a camada de GAP é utilizada após a saída da última camada convolucional, conectando-a à camada de classificação ou camada totalmente conectada.

Por exemplo, a saída da última camada convolucional da rede Xception possui as seguintes dimensões $10 \times 10 \times 2048$, ou seja, 2048 mapas de características de dimensões 10×10 *pixels*. Ao adicionar uma camada GAP, cada um desses 2048 mapas são resumidos pela média dos valores dos seus *pixels*, resultando em um vetor contendo 2048 características. Outra forma de gerar vetores de características é utilizando uma camada de *flatten*, como será explicado na seção seguinte.

2.3.7 Camada de Flatten

Flatten é uma transformação aplicada à saída de uma camada, transformando um tensor em um vetor de características para ser utilizado como entrada para as camadas totalmente conectadas (*fully connected layers*). É uma camada utilizada normalmente após as camadas convolucionais ou de pooling. Por exemplo, a saída de uma camada de *pooling* contendo 20 mapas de características de dimensões 3×3 *pixels*, resulta em um vetor contendo 180 características.

2.3.8 Arquiteturas de Redes Neurais Convolucionais

Um exemplo de uma arquitetura simples de uma CNN é a LeNet-5, composta por duas camadas convolucionais (filtros 5×5) seguidas por operações de *pooling* (filtros $2 \times 2 - stride 2$) e três camadas totalmente conectadas contendo 120, 84 e 10 neurônios, sendo a última a camada de saída. A LeNet-5 foi originalmente proposta para classificação de imagens de dígitos manuscritos. A Figura 14 apresenta sua arquitetura.

Muitas outras arquiteturas de CNN foram propostas nas últimas duas décadas, tornando a escolha da melhor arquitetura um grande desafio. Essas arquiteturas variam em diferentes aspectos como: número de camadas, tipos de camadas, tamanho dos filtros convolucionais, operações, conexões entre as camadas, entre outros aspectos (ELHAS-SOUNY; SMARANDACHE, 2019; KHAN, A. et al., 2020). Consequentemente, é importante testar diferentes arquiteturas e escolher aquela que melhor se adeque ao tipo de problema. Algumas arquiteturas amplamente utilizadas na literatura para a tarefa de classificação e diagnóstico por imagens são: VGG (SIMONYAN; ZISSERMAN, 2015), ResNet (HE et al., 2016), DenseNet (HUANG, G. et al., 2016), InceptionV3 (SZEGEDY; VANHOUCKE et al., 2016), InceptionResNetV2 (SZEGEDY; IOFFE et al., 2017), Xception (CHOLLET, 2017), MobileNet (HOWARD et al., 2017), MobileNetV2 (SANDLER et al., 2018) e EfficientNet (TAN; LE, 2019).



Figura 14: Arquitetura da LeNet-5. Adaptado de (LECUN et al., 1998).

2.4 Redes Neurais Recorrentes

As Redes Neurais Recorrentes (RNNs, *Recurrent Neural Networks*) foram introduzidas na década de 1980 (JORDAN, 1997; RUMELHART; HINTON; WILLIAMS, 1986) como redes especializadas em processamento de dados sequenciais (GOODFELLOW; BENGIO; COURVILLE, 2016). As RNNs são um tipo específico de ANNs inspiradas no conceito de memória biológica, onde a informação temporal é usada na computação e os pesos da rede são compartilhados ao longo do tempo, como uma forma de persistir informações.

As redes do tipo *feedforward* como as MLPs e as CNNs, apenas transmitem informações da camada de entrada para a camada de saída, assumindo que todas as entradas são independentes umas das outras, desconsiderando as dependências temporais. No entanto, para muitas aplicações (por exemplo, tradução de idiomas, reconhecimento de fala e reconhecimento de atividades humanas), as informações presentes nas sequências de dados são de grande importância e devem ser consideradas.

As RNNs podem trabalhar com diferentes configurações de entrada e saída, dependendo do objetivo a ser alcançado. Por exemplo, na classificação de vídeos o objetivo é receber uma sequência de *frames* e classificá-los. Portanto, espera-se que a RNN receba como entrada uma sequência de *frames* e devolva como saída o resultado da classificação (por exemplo, pneumonia). Para o caso apresentado, temos uma RNN do tipo *many-toone* (ou muitos para um), no entanto, essas relações podem se alterar, conforme a tarefa escolhida.

O termo recorrente (*recurrent*) nas RNNs advém do fato de que essas redes funcionam em *loop*. Os cálculos no tempo t dependem dos cálculos realizados no tempo t - 1. Essa dependência temporal é preservada em um estado, conhecido como estado oculto (h). A forma como a rede persiste informações se dá por meio do compartilhamento dos seus pesos ao longo do tempo. A Equação 2.16 define o estado oculto no tempo t em função de t - 1.

$$h_t = a(Ux_t + Wh_{t-1} + b) \tag{2.16}$$

Onde h_t é o estado oculto no tempo t, x_t é a entrada no tempo t ponderada pela matriz de pesos da entrada U, h_{t-1} é o estado oculto no tempo t-1 (memória da rede), W é a matriz de transição do estado oculto, b é o bias e a é uma função de ativação não linear (por exemplo, tanh). Caso o estado oculto h_{t-1} não tenha sido inicializado, adota-se zero.

A saída o_t relativa à entrada x_t será dada por:

$$o_t = Vh_t + c \tag{2.17}$$

Onde o_t representa a saída no tempo t, V é uma matriz de pesos que pondera a saída, h_t é o estado oculto no tempo t, e c é o *bias*. Aplica-se então uma função de ativação a saída o_t para a classificação, conforme a Equação 2.18:

$$\hat{y}_t = a(o_t) \tag{2.18}$$

Onde \hat{y}_t é a saída predita, a uma função de ativação (por exemplo, softmax) e o_t a saída no tempo t.

As matrizes de pesos W, $U \in V$ atuam no sentido de ponderar a importância, tanto da informação atual quanto do passado. Dessa forma, é possível ajustar os pesos na etapa de treinamento, considerando os erros obtidos em cada tempo t, conforme Equação 2.19.

$$L(y, \hat{y}) = \sum_{t} L_t(y_t, \hat{y}_t)$$
(2.19)

Onde L é uma função de perda, y_t é a saída esperada em cada etapa de tempo t e \hat{y}_t é a predição. A Figura 15 apresenta uma RNN desenvolada (*unfolded*) representando o fluxo de entrada e saída da rede.



Figura 15: Representação de uma RNN desenrolada. Adaptado de (GOODFELLOW; BENGIO; COURVILLE, 2016).

Embora esse tipo de ANN possua relevância para problemas sequenciais, as RNNs sofrem de um problema conhecido como desaparecimento do gradiente (*vanishing gradiente*), onde os gradientes de tempos mais distantes se dissipam parando de contribuir com o aprendizado da rede (HOCHREITER, 1998; SHERSTINSKY, 2018). Portanto, novas arquiteturas foram investigadas no intuito de solucionar esse problema, uma delas é a LSTM, apresentada na próxima seção.

2.4.1 Long Short-Term Memory

As redes do tipo LSTM (do inglês *Long Short-Term Memory*) foram introduzidas em Hochreiter e Schmidhuber (1997) e são um tipo específico de RNNs capazes de lidar com o problema do desaparecimento do gradiente, que torna o aprendizado de sequências muito longas uma tarefa difícil de ser realizada (HOCHREITER, 1998).

As LSTMs podem trabalhar com longas sequências, considerando informações em tempos mais distantes. Dessa forma, as LSTMs têm sido aplicadas em diferentes áreas do conhecimento, sendo frequentemente utilizadas na área de processamento de linguagem natural. Na área de VC, as LSTMs também foram utilizadas com sucesso, por exemplo, nas tarefas de reconhecimento e descrição de imagens e vídeos (DONAHUE et al., 2017).

As LSTMs usam o conceito de *gates*, que possuem o objetivo de regular o fluxo de informações na rede. Nesse sentido, os *gates* aprendem quais informações devem ser adicionadas, mantidas ou descartadas. Uma diferença em relação às RNNs é que as LSTMs possuem dois estados, o estado da célula (c) e o estado oculto (h). Eles carregam a memória de longo e curto prazo, respectivamente.

A Figura 16 apresenta dois diagramas, do lado esquerdo o diagrama de uma unidade RNN, do lado direito o diagrama de uma unidade LSTM contendo o estado oculto, o estado da célula e todos os *gates* que controlam o fluxo de informações (*forget, input, input modulation* e *output*).



Figura 16: Representação de unidades RNN e LSTM. Adaptado de (DONAHUE et al., 2017).

2.4.1.1 Estado da Célula

O estado da célula (c) é responsável por armazenar as informações de tempos anteriores, portanto, tem uma natureza recursiva (assim como o estado oculto h), como pode ser observada na Figura 16 pelas setas que entram e saem das áreas pontilhadas. O estado da célula é atualizado considerando o que deve ser esquecido do estado da célula no passo anterior e a nova informação que será modulada e adicionada. Sua definição é dada por:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{2.20}$$

Onde c_t é o estado da célula a ser atualizado, f_t é o forget gate, c_{t-1} é o estado da célula no passo anterior, i_t é o input gate, g_t é o input modulation gate, e $x \odot y$ denota a operação multiplicação elemento a elemento dos vetores $x \in y$. Cada um desses termos da equação serão apresentados nas próximas seções.

2.4.1.2 Forget Gate

O forget gate (f) é responsável por esquecer as informações que não são mais necessárias para o estado da célula. Dessa forma, recebe como entrada o estado oculto h_{t-1} do passo anterior e a entrada atual x_t , onde essas entradas são ponderadas por matrizes de pesos W_f e somadas a um bias b_f . É aplicada uma função de ativação sigmoide, que resulta em um número real entre 0 e 1 para cada valor do estado da célula anterior (c_{t-1}) . Nesse sentido, 0 indica que a informação deve ser completamente esquecida e 1 que a informação deve ser completamente mantida.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$
(2.21)

2.4.1.3 Input Gate e Input Modulation Gate

O *input gate* (*i*) é responsável por determinar quais informações devem ser adicionadas ao estado da célula (memória de longo prazo). Cada um desses *gates* utilizam uma função de ativação diferente. O *input gate* utiliza uma função *sigmoide* e é definido como:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.22}$$

O input modulation gate (g) utiliza a função de ativação tanh, conforme definido na

Equação 2.23 e atua como um modulador, gerando as informações candidatas que serão adicionadas ao estado da célula, com base no que é determinado pelo *input gate*.

$$g_t = tanh(W_c[h_{t-1}, x_t] + b_c)$$
(2.23)

2.4.1.4 Output Gate

O *output gate* é responsável por filtrar quais informações do estado da célula (c_t) serão propagadas pelo estado oculto (h_t) . Utiliza uma função de ativação *sigmoide*. Sua definição é dada por:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.24}$$

O estado oculto (h_t) é responsável por propagar as informações relevantes do estado da célula (c_t) , sendo definido por:

$$h_t = o_t \odot tanh(c_t) \tag{2.25}$$

Onde o_t é a saída do *output gate* e c_t é o estado atual da célula.

2.5 Transferência de Aprendizado

A transferência de aprendizado (*transfer learning*) é uma estratégia eficaz para treinar uma CNN (não se resumindo a esse tipo de rede) em um pequeno conjunto de dados. Nesta técnica, a CNN pré-treinada em um grande conjunto de dados, como o ImageNet (DENG et al., 2009), é reutilizada em uma tarefa similar, aproveitando as características já aprendidas (WANG, K. et al., 2020). O conjunto de dados ImageNet contém mais de 14 milhões de imagens e centenas de categorias de objetos anotados. Sua criação teve como objetivo facilitar o treinamento de algoritmos de aprendizado profundo com foco em classificação de imagens, localização de objetos e detecção de objetos (RUSSAKOVSKY et al., 2014). As próximas seções apresentam os métodos de transferência de aprendizado, geralmente utilizados em problemas de classificação de imagens.

2.5.1 Métodos de Transferência de Aprendizado

A transferência de aprendizado nas CNNs pode ser realizada utilizando dois métodos. O primeiro é conhecido como transferência de aprendizado sem ajuste fino e o segundo com ajuste fino, pois neste os pesos da camada convolucional sofrem ajuste. As Seções 2.5.1.1, 2.5.1.2, e 2.5.1.3 foram baseadas no guia de transferência de aprendizado apresentado na documentação do *framework Keras* (KERAS, 2020).

2.5.1.1 Método sem Ajuste Fino

No método sem ajuste fino, removem-se as camadas totalmente conectadas de um modelo pré-treinado, mantendo-se as camadas convolucionais congeladas (seus pesos não são alterados). Adiciona-se uma ou mais camadas densas após as camadas convolucionais para serem treinadas usando pequenos conjuntos de dados. Nesse processo, apenas os pesos das camadas recém-adicionadas são aprendidos, aproveitando os pesos congelados das camadas convolucionais, reduzindo consideravelmente o número de parâmetros a serem aprendidos.

2.5.1.2 Método com Ajuste Fino

No método com ajuste fino, o procedimento é parecido com o anterior, no entanto, toda ou parte das camadas convolucionais são descongeladas e o treinamento da rede é realizado com uma taxa de aprendizado pequena, onde são utilizados os pesos iniciais fornecidos pela CNN pré-treinada, ajustando o aprendizado ao novo conjunto de dados.

2.5.1.3 Extração de Características

A extração de características é uma etapa importante para o reconhecimento de padrões. As CNNs podem tanto extrair essas características das imagens, utilizando as camadas convolucionais (Seção 2.3.4), quanto classificá-las com base nessas características, utilizando suas camadas totalmente conectadas.

Para que uma CNN possa extrair características de uma imagem, é necessário inicialmente realizar um treinamento em um conjunto de dados de interesse. Dessa forma, a CNN consegue aprender os valores dos coeficientes dos filtros convolucionais capazes de realçar determinadas características das imagens, aquelas mais relevantes ao problema considerado. Nesse sentido, TL pode ser utilizado para acelerar o processo de aprendizado, transferindo para as novas redes as características aprendidas em outro conjunto de dados.

Na extração de características, as camadas densas de uma CNN pré-treinada são removidas e as camadas convolucionais são preservadas. Dessa forma, a parte preservada da rede funciona como um extrator de características espaciais, servindo como entrada para outros tipos de redes, por exemplo, as RNNs.

A Figura 17 apresenta um exemplo da extração de características utilizando uma VGG16. A Figura 17A representa uma imagem de entrada de dimensões $224 \times 224 \times 3$ *pixels*, a Figura 17B representa as camadas convolucionais, com *max pooling* e *flatten*. Por fim, a Figura 17C representa a saída da camada de *flatten*, um vetor com 25.088 características.



Figura 17: Exemplo da extração de características utilizando uma VGG16.

2.6 Sobreajuste

O sobreajuste (*overfitting*) ocorre quando uma rede aprende perfeitamente os pesos com base no conjunto de treinamento, mas não consegue capturar adequadamente o processo que os gerou, ou seja, não conseguem generalizar. Portanto, o erro fora do conjunto de treinamento (exemplos do conjunto de dados que a rede não conhece) é grande se comparado ao erro do treinamento (exemplos do conjunto de dados que a rede conhece), apresentando uma grande variação.

Em aprendizado profundo, o sobreajuste pode ocorrer nos casos onde o conjunto de treinamento não possua dados suficientes e, consequentemente, o modelo preditivo não pode generalizar e prever com sucesso instâncias não vistas do problema na etapa de treinamento. Para melhorar a generalização nas redes profundas, é necessário aumentar a quantidade de dados, reduzir a complexidade do modelo ou aplicar técnicas de regularização. Nesse sentido, a Seção 2.6.1 apresenta a camada de *dropout*, cujo objetivo é combater o sobreajuste.

2.6.1 Camada de Dropout

A regularização é uma técnica cujo objetivo é reduzir o sobreajuste causado por modelos preditivos muito complexos, que acabam se ajustando perfeitamente aos dados. Uma forma de reduzir a complexidade desses modelos é por meio da regularização. Dessa forma, normalmente adiciona-se um termo de regularização (norma L1 e L2) à função de custo, para penalizar parâmetros muito grandes, gerando modelos mais simples.

A camada de *dropout* (SRIVASTAVA et al., 2014) atua nesse sentido, porém possui um funcionamento um pouco diferente da técnica de regularização L1 e L2. No *dropout* a regularização é realizada diretamente na arquitetura da rede, de tal forma que alguns neurônios e suas conexões são descartados com uma probabilidade p (normalmente entre 0,1 e 0,5) durante o treinamento, contribuindo para que os neurônios não se adaptem demais.

2.7 Validação Cruzada

A validação cruzada é uma técnica de reamostragem de dados utilizada para a avaliação da capacidade de generalização dos modelos preditivos e para prevenir o sobreajuste (BERRAR, 2019). Diferentes métodos podem ser utilizados para fazer a reamostragem de dados, porém o método normalmente utilizado é conhecido como K-fold (ou k-partições).

Nesse método o conjunto de dados é particionado em k partições disjuntas, de aproximadamente mesmo tamanho. Com base nesse particionamento, o modelo é treinado e testado k vezes, onde a cada iteração o treinamento é realizado em k - 1 partições (conjunto de treinamento) e testado na partição restante (conjunto de teste), sem que as k partições de teste se repitam.

O particionamento dos dados é realizado por amostragem aleatória sem substituição. As k partições são estratificadas conforme a distribuição das classes no conjunto de dados, garantindo que cada partição possua aproximadamente a mesma distribuição em relação ao conjunto original. No entanto, é preciso definir um valor para k, onde k representa a quantidade de partições que deverão ser geradas e, consequentemente, a quantidade de iterações que serão realizadas. A Figura 18 apresenta o particionamento em k partições. Alguns estudos descrevem formas de se chegar a um valor ótimo para k. Em Kohavi et al. (1995), sugere-se um valor fixo de k = 5, buscando-se um melhor equilíbrio entre viés, variância e custo computacional. Em Anguita et al. (2012), a proposta foi utilizar o valor de k como um hiperparâmetro, portanto, deve ser otimizado durante a fase de seleção dos modelos. Recentemente em Marcot e Hanea (2021) foi testado diferentes valores para kcom modelos baseados em redes bayesianas; foi sugerido o uso de k = 10, sendo suficiente em alguns casos adotar o valor de k = 5, a depender da quantidade de dados disponíveis.

O desempenho final do modelo é expresso pela média do desempenho em cada uma das k partições.

	Conjunto de Dados				
lteração 1	Teste	Treinamento	Treinamento	Treinamento	Treinamento
lteração 2	Treinamento	Teste	Treinamento	Treinamento	Treinamento
Iteração 3	Treinamento	Treinamento	Teste	Treinamento	Treinamento
Iteração k-1	Treinamento	Treinamento	Treinamento	Teste	Treinamento
lteração k	Treinamento	Treinamento	Treinamento	Treinamento	Teste
	Partição 1	Partição 2	Partição 3	Partição k-1	Partição k

Figura 18: Validação cruzada pelo método K-fold com k partições.

2.8 Funções de Ativação

A saída de um neurônio é dada por uma função de ativação. As funções de ativação utilizadas nas redes profundas são funções não lineares e diferenciáveis, permitindo o treinamento por *backpropagation*, conforme explicado na Seção 2.2.4. Essas funções são utilizadas nas camadas ocultas com a finalidade de permitir que funções mais complexas possam ser combinadas ao longo da rede. Dentre elas, as mais utilizadas são as apresentadas nas próximas seções.

2.8.1 Sigmoide

A função sigmoide também conhecida como função logística, recebe um número real como entrada (z) e retorna um número real no intervalo [0,1] como saída. É a função de ativação utilizada para a classificação binária, onde a saída representa a probabilidade da entrada pertencer a uma determinada classe. A função é definida por:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.26}$$

2.8.2 Softmax

A função softmax também conhecida como softargmax (ou função exponencial normalizada), recebe um vetor (z) contendo k valores reais e retorna um vetor contendo k valores reais no invervalo [0,1] que somam 1. Essa função é utilizada na última camada da rede para problemas de classificação multiclasse, onde a saída representa a probabilidade da entrada pertencer a cada uma das classes. A função é definida por:

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}$$
 (2.27)

2.8.3 Tanh

A função tanh (do inglês Hyperbolic Tangent) recebe um número real como entrada (z) e retorna um número real no intervalo [-1, 1]. Essa função é utilizada nas camadas intermediárias ou camadas ocultas da rede. A função é definida por:

$$tanh(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$$
(2.28)

2.8.4 ReLU

ReLU (do inglês *Rectified Linear Function*) recebe um número real e retém somente a parte positiva, descartando a parte negativa, que assume o valor de 0. Essa função é utilizada nas camadas intermediárias ou camadas ocultas da rede. Por ser uma transformação não linear muito simples, possui também vantagens quanto ao desempenho. A função é definida por:

$$ReLU(z) = \begin{cases} z, se \ z > 0\\ 0, cc. \end{cases}$$
(2.29)

2.9 Função de Perda

Uma função de perda (*loss function*), também conhecida como função de custo (*cost function*), mede a discrepância entre as predições de saída da rede e os valores dos rótulos esperados (*ground truth*). A função de perda normalmente utilizada para problemas de classificação é a função de entropia cruzada. A função do erro quadrático médio (MSE, *Mean Square Error*) é normalmente a função utilizada em tarefas de regressão. Dessa forma, a função de perda possui um papel importante no treinamento da rede e ela é definida segundo a tarefa a ser desempenhada.

2.10 Otimização de Hiperparâmetros

O problema de otimização de hiperparâmetros (HPO, *Hyperparameter Optimization*) pode ser caracterizado pela busca do melhor conjunto de hiperparâmetros que maximize ou minimize o resultado de uma função objetivo.

Cada hiperparâmetro faz parte de um aspecto da configuração de um modelo preditivo, sendo uma dimensão do espaço de busca. Diferentemente de um parâmetro (peso) de uma rede neural, um hiperparâmetro não é aprendido no processo de treinamento (*backpropagation*). Dessa forma, testar todas as configurações de hiperparâmetros pode levar um tempo considerável, dada a grande quantidade de hiperparâmetros normalmente disponíveis para esses tipos de modelos preditivos. (BERGSTRA; BENGIO, 2012).

Os dois métodos mais utilizados são o *Grid Search* e o *Random Search*. No *Grid Search*, a busca é realizada exaustivamente usando todas as combinações de hiperparâmetros do espaço de busca. No *Random Search*, a busca é realizada de forma aleatória, portanto, com menor custo computacional do que a busca realizada no primeiro método, porém sem garantia de um resultado ótimo.

Os hiperparâmetros influenciam no resultado do treinamento e, consequentemente, nas medidas de avaliação, por exemplo, a acurácia. Porém, devido à dificuldade de realizar a otimização considerando a busca por toda a grade de busca (*Grid Search*), surgiram outros métodos, principalmente para tornar o processo cada vez mais eficiente.

A Otimização Bayesiana é um método conhecido como caixa-preta (*black-box*) baseado em um modelo de probabilidade que utiliza as obervações anteriores para aprender uma função objetivo custosa (FEURER; HUTTER, 2019). Algumas implementações conhecidas são: SMAC (HUTTER; HOOS; LEYTON-BROWN, 2011), HYPEROPTO (BERGSTRA; YAMINS; COX, 2013), MOE (CLARK et al., 2014) e pyGPGO (JIMÉ-NEZ; GINEBRA, 2017). Ao contrário da otimização Bayesiana, os métodos conhecidos como *Multi Fidelity* buscam estimar a função objetivo de uma maneira mais barata. Dentre esses métodos, pode-se destacar: o *Successive Halving* (JAMIESON; TALWALKAR, 2015), o *Hyperband* (LI et al., 2017) e o *BOHB* (FALKNER; KLEIN; HUTTER, 2018).

O método de *Successive Halving* prioriza os modelos preditivos que são mais promissores, fornecendo mais recursos (tempo) a esses modelos. A cada iteração são mantidas metade das melhores configurações de hiperparâmetros e descartados metade dos modelos que não obtiveram um bom desempenho. Esse procedimento se repete até que só reste apenas uma configuração de hiperparâmetros. No entanto, nesse método é necessário definir quantos conjuntos de hiperparâmetros serão selecionados e quantas iterações serão realizadas no início. Dessa forma, o *Hyperband* é uma proposta para estender o método de *Successive Halving* e visa solucionar este problema, realizando frequentemente o método de *Succesive Halving* com diferentes recursos (LI et al., 2017). O método *BOHB* (FALKNER; KLEIN; HUTTER, 2018) é uma combinação eficiente entre o *Hyperband* e a Otimização Bayesiana.

2.11 Avaliação dos Classificadores

Para que seja possível quantificar, avaliar e comparar o desempenho de diferentes classificadores, utilizam-se algumas medidas de avaliação (ou na área médica, medidas clínicas de diagnóstico (FERREIRA; PATINO, 2017)). Essas medidas variam em importância dependendo do problema, porém elas se baseiam em algum aspecto da matriz de confusão.

A matriz de confusão ou tabela de contingência é uma tabela utilizada para organizar as decisões tomadas pelo classificador, tornando explícito como cada classe está sendo confundida com a outra (PROVOST; FAWCETT, 2013). A matriz de confusão possui o número de linhas e colunas iguais ao número de classes. As linhas representam o resultado predito pelo classificador e as colunas o valor verdadeiro. Cada célula da tabela representa uma contagem que serve como base para a composição de outras medidas, por exemplo, a acurácia. A Figura 19 apresenta a estrutura da matriz de confusão.

Conforme a Figura 19, as seguintes observações podem ser realizadas: os verdadeiros positivos (VP) são os casos positivos (P) classificados corretamente; os falsos positivos (FP) são os casos negativos (N) classificados incorretamente como positivos; os verdadeiros negativos (VN) são os casos negativos classificados corretamente; e os falsos negativos



Figura 19: Estrutura da matriz de confusão.

(FN) são os casos positivos classificados incorretamente como negativos.

As principais medidas utilizadas na avaliação de classificadores são apresentadas nas seções seguintes.

2.11.1 Acurácia

Acurácia é frequentemente utilizada para a comparação de classificadores e representa a taxa de acerto, ou seja, o quão próximo é o valor predito se comparado ao valor verdadeiro. A acurácia é definida pela equação:

$$Acurácia = \frac{VP + VN}{P + N} = \frac{VP + VN}{VP + VN + FP + FN}$$
(2.30)

2.11.2 Precisão

Precisão ou valor preditivo positivo (VPP) é o número de casos positivos classificados corretamente, dentre todos os casos classificados como positivos. A precisão é definida pela equação:

$$Precisão = \frac{VP}{VP + FP} \tag{2.31}$$

2.11.3 Sensibilidade

Sensibilidade ou revocação (*recall*) é a taxa de verdadeiros positivos (TVP), ou seja, os casos positivos classificados corretamente, dentre os casos verdadeiramente positivos. A sensibilidade é definida pela equação:

$$Sensibilidade = \frac{VP}{P} = \frac{VP}{VP + FN}$$
(2.32)

2.11.4 Especificidade

Especificidade é a taxa de verdadeiros negativos (TVN), ou seja, os casos negativos classificados corretamente, dentre os casos verdadeiramente negativos. A especificidade é definida pela equação:

$$Especificidade = \frac{VN}{N} = \frac{VN}{VN + FP}$$
(2.33)

2.11.5 F1-Score

F1-Score é calculado com base na média harmônica entre a precisão e a sensibilidade. O F1-Score é definido na equação:

$$F1_{Score} = 2 \times \frac{Precisão \times Sensibilidade}{Precisão + Sensibilidade}$$
(2.34)

2.11.6 Coeficiente de Correlação

O coeficiente de correlação é uma estatística usada para medir a associação entre duas variáveis quantitativas. Ele mede como essas duas variáveis variam conjuntamente, indicando se existe ou não um relacionamento entre elas (MORETTIN; BUSSAB, 2017). Adicionalmente, é possível analisar visualmente o relacionamento entre as duas variáveis através de um gráfico de dispersão (*scatter plot*). Cada ponto (x,y) no gráfico cartesiano representa um exemplo do conjunto de dados, x assume o papel da variável independente e y o da variável dependente, porém não é necessário que exista uma relação de causa. O comportamento desses pontos no gráfico permite avaliar a existência de um relacionamento entre as variáveis (CAPP; NIENOV, 2020). Dessa forma, normalmente utiliza-se a análise visual em conjunto com um teste de correlação.

O coeficiente de correlação varia entre -1 (correlação negativa perfeita) e 1 (correlação positiva perfeita). Uma correlação positiva indica que x e y variam no mesmo sentido (quando x aumenta, y aumenta) e uma correlação negativa o inverso (quando x aumenta, y diminui). O valor do coeficiente pode ser analisado de maneira qualitativa. A Tabela

1 apresenta a avaliação qualitativa com base em um intervalo de valores que o coeficiente de correlação pode assumir (CAPP; NIENOV, 2020).

Valor do Coeficiente $(+/-)$	Intensidade da Correlação
0.000	Nula
0,001 a 0,299	Fraca
$0,300 \ a \ 0,599$	Moderada
0,600 a 0,899	Forte
0.900	Muito forte
1	Perfeita

Tabela 1: Avaliação qualitativa do coeficiente de correlação.

Existem alguns testes de correlação que podem ser utilizados. A escolha desses testes depende do tipo das variáveis consideradas (variáveis quantitativas normais, quantitativas não normais e qualitativas ordinais). O teste de correlação de Pearson (r) pressupõe variáveis quantitativas com uma distribuição normal (teste paramétrico), sendo afetado por valores extremos (*outliers*). O teste de Spearman (ρ) assume variáveis quantitativas não normais (não paramétrico) e qualitativas ordinárias, não sendo afetado por valores extremos. O teste de Kendall (τ) possui as vantagens do teste de Spearman, porém pode ser utilizado com um conjunto de dados pequeno e com um grande número de postos empatados (empate na ordenação dos pares de uma amostra), conforme será visto na próxima seção (CAPP; NIENOV, 2020).

2.11.6.1 Coeficiente de Correlação de Kendall

O coeficiente de correlação de Kendall foi proposto por Maurice Kendall em 1938 (KEN-DALL, 1938) e permite o uso de variáveis qualitativas ordinais. É representado pela letra grega tau (τ), podendo ser utilizado nos casos onde existam muitos postos empatados (ajuste conhecido como tau-b).

O teste de correlação testa se uma amostra coletada de uma população possui ou não correlação entre as duas variáveis observadas. Nesse sentido, o teste de correlação tau de Kendall pode ser utilizado obtendo-se a distribuição amostral da estatística τ definida pela equação:

$$\tau = \frac{n_c - n_d}{n_p} \tag{2.35}$$

Onde n_c é o número de pares concordantes, n_d o número de pares discordantes e n_p

o número total de pares. Um par $\{(x_i, y_i), (x_j, y_j)\}$ é considerado concordante se: $x_i > x_j$ e $y_i > y_j$ ou $x_i < x_j$ e $y_i < y_j$; discordante se $x_i > x_j$ e $y_i < y_j$ ou $x_i < x_j$ e $y_i > y_j$; empatado se $x_i = x_j$ ou $y_i = y_j$.

O número total de pares é dado por:

$$n_p = n(n-1)/2 \tag{2.36}$$

Onde n é o número de exemplos da amostra.

Para o caso de postos empatados utiliza-se o ajuste definido pelas equações:

$$\tau_B = \frac{n_c - n_d}{\sqrt{(n_p - n_x)(n_p - n_y)}}$$
(2.37)

$$n_x = \sum_i t_i (t_i - 1)/2 \tag{2.38}$$

$$n_y = \sum_i u_j (u_j - 1)/2 \tag{2.39}$$

Onde t_i é o número de elementos no *i*-ésimo grupo de empates entre os valores de x, e u_j o número de elementos no *j*-ésimo grupo de empates entre os valores de y.

2.11.6.2 Construção do Teste de Hipóteses

O teste de hipóteses é construído da seguinte maneira (quando não se sabe a natureza do relacionamento): a hipótese nula é definida por $H_0: \tau_B(X,Y) = 0$ (não existe correlação entre as variáveis) e a hipótese alternativa por $H_1: \tau_B(X,Y) \neq 0$. Para que exista correlação entre as variáveis aleatórias X e Y é necessário rejeitar H_0 no nível de significância ou risco que se deseja assumir (por exemplo, 5% de probabilidade de rejeição da hipótese nula quando ela é verdadeira), e que preferencialmente o coeficiente de correlação apresente um valor significativo, conforme avaliação qualitativa apresentada na Tabela 1.

3 Trabalhos Relacionados

Este capítulo apresenta os trabalhos que utilizam técnicas de aprendizado profundo e imagens de USP para o diagnóstico de doenças pulmonares e de COVID-19. Um resumo dos trabalhos é apresentado na Tabela 18. Na Seção 3.1, são apresentados os trabalhos relacionados às doenças pulmonares, que servem de alicerce para aqueles apresentados na Seção 3.2, onde o foco está no diagnóstico da COVID-19. Na Seção 3.3 são apresentadas as conclusões sobre os trabalhos relacionados. Por fim, a Seção 3.4 apresenta como o método proposto nesta dissertação se diferencia dos demais trabalhos.

3.1 Doenças Pulmonares

Esta seção é dedicada aos trabalhos que utilizaram técnicas de aprendizado profundo e imagens de USP para o diagnóstico de doenças pulmonares em um contexto mais amplo. As próximas seções estão organizadas em ordem cronológica, onde cada seção apresenta um trabalho relacionado.

3.1.1 Kulhare et al. (2018)

Em Kulhare et al. (2018), os autores propuseram o uso de um detector de disparo único (SSD, *Single Shot Detector*), uma extensão das CNNs baseadas em região (R-CNN, *Region Based Convolutional Neural Networks*), com arquitetura Inception V2, para detectar linhas B, linhas B coalescentes, consolidações e derrame pleural em nível de *frame*. Também foi proposto o uso de uma CNN, com arquitetura Inception V3, para classificar a perda do deslizamento pulmonar (classificação binária).

O conjunto de dados foi composto por 2.200 vídeos curtos de suínos (*in vivo*), contendo 100 exames de USP adquiridos por um transdutor convexo e anotados por radiologistas e ultrassonografistas especializados. Foram utilizadas duas abordagens para a construção do conjunto de dados. A primeira foi utilizada para detecção das linhas A, linhas B, linhas B coalescentes, consolidações, derrame pleural e linha pleural. A segunda foi utilizada para gerar uma representação da doença pneumotórax, esta relacionada à ausência do deslizamento pulmonar (diagnosticada no modo movimento ou modo M).

Segundo os autores, os dados brutos capturados por transdutores convexos assumem a forma de coordenadas polares, dessa forma os vídeos foram transformados para coordenadas cartesianas, servindo para eliminar a variação angular das linhas B, acelerando o aprendizado. Após essa transformação de coordenadas, foram realizados alguns préprocessamentos.

Na primeira abordagem, os *frames* extraídos dos vídeos foram recortados para remover informações desnecessárias (bordas escuras e textos), resultando em *frames* com uma resolução de 801×555 *pixels*. Na segunda abordagem, uma representação da doença pneumotórax foi gerada utilizando os vídeos em um modo movimento (modo M) simulado, onde foi possível realizar a reconstrução do movimento e extrair os *frames* referentes ao modo M.

Foi utilizada a técnica de *data augmentation*. Para os *frames* referentes aos achados como linhas A, linhas B, linhas B coalescentes, consolidações, derrame pleural e linha pleural, foi realizado o seguinte pré-processamento: inversão horizontal, cortes aleatórios (*random cropping*), escala e translação. Para os *frames* referentes ao deslizamento pulmonar, foi realizado o seguinte pré-processamento: desfoque gaussiano (*gaussian blur*) e variações no contraste e brilho.

Ambas as redes foram inicializadas com os pesos pré-treinados no conjunto de dados ImageNet. No entanto, somente a rede SSD recebeu um ajuste fino nas camadas convolucionais. Para a CNN (Inception V3) foi realizado apenas o treinamento das camadas de classificação. Os seguintes hiperparâmetros foram reportados para o treinamento do SSD: 300.000 épocas, tamanho do lote de 24, momentum de 0,9 e uma taxa de aprendizado de 1×10^{-4} decaindo a cada 80.000 iterações com uma taxa de 0,95. Para o treinamento da CNN, foram reportados os hiperparâmetros: 10.000 épocas, tamanho do lote de 100 e taxa de aprendizado de 1×10^{-3} constante. A distribuição de *frames* no conjunto de treinamento e teste variou em relação aos achados radiológicos, no entanto, não foi informada a proporção exata.

Os resultados indicaram uma acurácia superior a 85%, pelo menos 85% de sensibilidade (exceto para as linhas B (28%)) e 86% de especificidade para todos os achados, demonstrando que o uso de modelos suínos pode ser um passo importante em direção aos ensaios clínicos com pacientes. Além disso, demonstraram que as CNNs podem auxiliar um operador com habilidade limitada a identificar esses achados nos exames de USP.

3.1.2 Sloun e Demi (2019)

No trabalho apresentado em Sloun e Demi (2019), os autores descreveram a arquitetura de uma CNN para a classificação da presença de linhas B em nível de *frame*.

Foram consideradas dois tipos de imagens: *in-vitro* e *in-vivo*. A imagem *in-vitro* foi fornecida por um simulador, capaz de mimetizar uma condição patológica, como a redução da aeração dos pulmões pela presença induzida de líquido, por exemplo. Essas simulações geraram as linhas B, adquiridas utilizando um transdutor linear com frequência de 4–5 MHz no modo B. Dessa forma, foram gerados 10 vídeos de USP, que totalizaram 3.162 *frames*. Esses *frames* foram anotados por um clínico especialista em USP com 20 anos de experiência, que os classificou em duas classes: 1 para a presença de linhas B e 0 para ausência.

O conjunto de treinamento *in vitro* foi composto por 6 vídeos e o conjunto de teste por 4 vídeos. Todos os *frames* das aquisições em modo B foram redimensionados para 256×256 *pixels*. Adicionalmente, foi realizada uma normalização tonal dos *pixels* (0–1).

Para os dados *in-vivo* foram adquiridos (mesmo transdutor e modo) um total de 15 vídeos de US referentes a 10 pacientes, totalizando 1.552 *frames*. Esses *frames* foram anotados da mesma forma que os dados *in-vitro*. O conjunto de treinamento (*in-vivo*) foi composto por 9 vídeos referentes a 6 pacientes e para o conjunto de teste foram usados 6 vídeos referentes a 4 pacientes. O conjunto de treinamento (*in-vivo*) foi composto por 9 vídeos referentes a 6 pacientes e para o conjunto de teste foram usados 6 vídeos referentes a 6 pacientes e para o conjunto de teste foram usados 6 vídeos referentes a 4 pacientes. O mesmo pré-processamento foi utilizado para os *frames* dos vídeos.

Além desses vídeos, foram adquiridos outros 12 vídeos do mesmo grupo de pacientes mencionado acima, porém utilizando um sistema comercial de auxílio à aquisição de vídeos e de exame de US da Toshiba (*The Aplio XV*) e um transdutor linear. Ao todo foram gerados 4.218 *frames*. Desse total, 7 vídeos foram usados para treinamento e 5 para teste. Todos os vídeos foram adquiridos utilizando o modo B e os *frames* extraídos receberam o mesmo pré-processamento dos outros dados.

A arquitetura da CNN proposta pelos autores foi composta por 6 blocos convolucionais, cada um compreendendo duas camadas convolucionais e uma operação de *max pooling* (2×2) . As camadas convolucionais utilizaram uma função de ativação não linear (ReLU). A saída das camadas convolucionais serviram de entrada para uma camada de GAP, transformando a saída em um vetor de características. Este foi utilizado como entrada para as camadas de classificação, contendo duas camadas totalmente conectadas, onde a saída utilizou uma função de ativação *softmax*, responsável pela classificação.

Foi utilizada a técnica de data augmentation com o seguinte pré-processamento: deformações, cortes (cropping), borramento (blur), variações no contraste e adição de ruído branco gaussiano (white gaussian noise). Para regularização da rede, foram adicionadas camadas de dropout com uma taxa de 0,3 para as camadas convolucionais e 0,5 para as camadas densas. O treinamento foi realizado utilizando a função de perda de entropia cruzada, um tamanho do lote (batch size) de 64, com o otimizador Adam (do inglês Adaptive Moment Estimation) configurado com uma taxa de aprendizado de 1×10^{-3} .

Os resultados foram avaliados em termos da acurácia, sensibilidade, especificidade, valor preditivo negativo e valor preditivo positivo. Para os dados *in-vitro* foram reportados os seguintes valores: 91,7% (acurácia), 91,5% (sensibilidade), 91,8% (especificidade), 95,0% (valor preditivo negativo) e 86,4% (valor preditivo positivo). Para os dados *in-vivo*: 83,9% (acurácia), 78,6% (sensibilidade), 86,8% (especificidade), 88,2% (valor preditivo negativo) e 76,3% (valor preditivo positivo). Por fim, para os dados da Toshiba (*in-vivo*): 89,2% (acurácia), 87,1% (sensibilidade), 93,0% (especificidade), 79,8% (valor preditivo negativo) e 95,8% (valor preditivo positivo). Em geral, os resultados *in vitro* foram mais significativos do que os *in vivo*.

3.1.3 Baloescu et al. (2020)

Em Baloescu et al. (2020) foram propostas duas arquiteturas customizadas de CNNs 3D em nível de vídeo, uma para classificar a presença das linhas B e outra para classificar o acometimento pulmonar com base numa pontuação que varia de 0–4, sendo: 0 para a ausência de linhas B (normal); 1 para uma linha B ocasional (normal); 2 para a presença de algumas linhas B (anormal); 3 para a presença de muitas linhas B (anormal); e 4 para a presença de incontáveis linhas B (quadro mais grave).

Foram coletados 400 vídeos da Emergência do Hospital Yale-New Haven, cada um pertencente a um único paciente. Os vídeos possuem uma duração média de 2,6 segundos com um frame rate variando entre 20 a 48 frames por segundo. Os 400 vídeos foram subdivididos em pequenas partes contendo 12 frames consecutivos, gerando um total de 2.415 vídeos para a análise. Todos esses vídeos foram anotados por dois ultrassonografistas experientes com base na pontuação mencionada anteriormente.

Os vídeos foram capturados com diferentes tipos de transdutores (linear, convexo e *phased array*). Para padronizar esses diferentes formatos de aquisição, os autores sugeriram a realização da padronização para um formato retilíneo consistente, onde as linhas B se apresentam alinhadas com a direção do feixe de ultrassom. Nesta operação também foram removidas informações desnecessárias (textos) referente aos 12 *frames* de cada vídeo.

Para o treinamento, foram considerados os dados de 300 (1.847 vídeos) dos 400 pacientes. Desses 300 pacientes, 85% dos dados foram usados para o treinamento da rede e 15% para a validação. O restante dos dados foram usados para a etapa de testes (100 vídeos), com a finalidade de manter a independência da amostra.

Foi utilizada a técnica de *data augmentation* com o seguinte pré-processamento: inversão horizontal, reversão dos *frames* no tempo, alterações na proporção do vídeo, pequenas rotações e variações no contraste. Todos os *frames* foram redimensionados para a resolução de 75×75 *pixels* e sofreram uma normalização tonal dos *pixels* (0–1).

Os autores propuseram uma arquitetura de CNN 3D customizada, consistindo de 8 camadas convolucionais intermediárias, seguidas por duas camadas totalmente conectadas. Foi utilizada a função de ativação não linear ReLU. A cada duas camadas intermediárias foram utilizadas as seguintes configurações: *stride* (1,1,1) com uma taxa de *dropout* de 0,1 e *stride* (2,2,1) com uma taxa de *dropout* de 0,2 e ao final da última camada intermediária foi utilizada uma camada de GAP. A primeira camada totalmente conectada foi configurada com 256 neurônios e a segunda com 512 neurônios. Na camada de classificação foram utilizados dois tipos de função ativação, onde a função *sigmoide* foi adotada para classificação da presença das linhas B e a função *softmax* para a pontuação da gravidade da doença. O total de parâmetros das duas redes foram de aproximadamente 4M de parâmetros.

As CNNs foram treinadas separadamente para cada tipo de tarefa utilizando como entrada os 12 *frames* de cada vídeo. As configurações de hiperparâmetros foram otimizadas considerando os resultados obtidos no conjunto de validação. As CNNs foram treinadas utilizando um tamanho do lote de 32, com o optimizador RMSProp configurado com uma taxa de aprendizado de 1×10^{-4} , decaindo a cada 500 iterações com uma taxa exponencial de 0,5. Foi adicionada uma parada precoce (*early stopping*) com base no desempenho da rede para prevenir o sobreajuste.

A CNN 3D proposta para a classificação binária atingiu uma sensibilidade de 93%, especificidade de 96%, AUC (do inglês *Area Under ROC Curve*) de 97%, e *Kappa* de 88% em concordância com especialistas humanos. Para a classificação multiclasse, o *Kappa* foi de 65%, indicado que a CNN 3D proposta para a classificação binária se ajustou melhor, possuindo potencial para auxiliar no diagnóstico de pacientes com queixas respiratórias, visto que o classificador apresentou uma concordância considerada forte (MCHUGH, 2012).

3.2 COVID-19

Esta seção é dedicada aos trabalhos que utilizaram técnicas de aprendizado profundo e imagens de USP para o diagnóstico da COVID-19. As próximas seções apresentam os trabalhos relacionados em ordem cronológica, agrupados por conjunto de dados.

3.2.1 ICLUS-DB: Italian COVID-19 Lung US Database

Os trabalhos organizados nesta seção utilizaram o conjunto de dados ICLUS-DB (do inglês *Italian COVID-19 Lung US Database*), disponível em: https://www.disi.unitn. it/iclus (acessado em 18 de dezembro de 2021). No entanto, o trabalho apresentado na Seção 3.2.1.1 utilizou uma versão estendida desse conjunto de dados, não disponível publicamente.

3.2.1.1 Roy et al. (2020)

Em Roy et al. (2020) foi proposta uma arquitetura de STN (do inglês *Spatial Transformer Network*) (JADERBERG; SIMONYAN; ZISSERMAN et al., 2015) para classificação em nível de *frame* e vídeo, para diagnosticar a gravidade do acometimento pulmonar causado pela COVID-19 com base numa pontuação que varia de 0–3 (SOLDATI et al., 2020).

Os autores utilizaram uma versão estendida completamente anotada do conjunto de dados ICLUS-DB, composto por um total de 277 vídeos de USP referentes a 35 pacientes, adquiridos em centros clínicos na Itália por diferentes tipos de transdutores.

O conjunto de dados para a classificação em nível de *frames* foi composto por um total de 58.924 *frames*, onde 45.560 foram adquiridos por transdutores convexos e 13.364 por transdutores lineares. A anotação dos dados em nível de *frames* seguiu um processo realizado em 4 níveis. No primeiro nível, quatro estudantes de mestrado com experiência em USP associaram aos *frames* os diagnósticos de gravidade. No segundo nível, a validação foi realizada por um estudante de doutorado com experiência em USP. No terceiro nível, a validação foi realizada por um engenheiro biomédico com mais de 10 anos de experiência

em USP. Finalmente, no quarto nível, a validação e a concordância foi realizada entre clínicos com mais de 10 anos de experiência em USP.

O conjunto de dados para classificação em nível de vídeo foi composto por 60 vídeos referentes a 35 pacientes. A anotação foi feita com base na concordância das previsões realizadas por 5 clínicos.

Para a classificação em nível de *frame*, os autores apresentaram uma nova arquitetura que eles chamaram de Reg-STN (do inglês *Regularised Spatial Transformer Networks*). Nessa arquitetura, a STN aprendeu duas transformações que foram aplicadas ao *frame* de entrada gerando duas imagens transformadas. Os resultados das transformações podem ser interpretados como as localizações dos achados que identificam um determinado diagnóstico. As imagens transformadas foram submetidas à duas CNNs, onde o treinamento foi feito utilizando uma função de perda conhecida como MSE (do inglês *Mean Squared Error*). O objetivo dessa função foi garantir a consistência entre as saídas (*logits*) das CNNs. Por fim, a classificação do diagnóstico foi realizada com base em uma das imagens transformadas. O treinamento final foi realizado utilizando uma função de perda que os autores chamaram de SORD (do inglês *Soft ORDinal Regression*). Essa função serviu para ponderar os rótulos categóricos ordinais, penalizando as predições dos diagnósticos mais distantes da verdade em contraste as pontuações mais próximas.

Para a classificação em nível de vídeo, os autores utilizaram uma estratégia que combina as classificações em nível de *frame* usando uma camada de agregação parametrizável baseada em uninormas (YAGER; RYBALOV, 1996). A camada de uninormas funciona como uma forma de suavizar uma regra mais rígida ao adotar, por exemplo, a pontuação máxima do diagnóstico referente a cada uma das classificações dos *frames*. Dessa forma, a camada de agregação proposta, recebeu como entrada uma sequência de diagnósticos referentes às classificações em nível de *frame* e agregou-os usando uma camada uninorma. A função *softmax* foi aplicada ao resultado da agregação que teve como saída a classificação em nível de vídeo.

O treinamento foi realizado separando a base de dados ICLUS-DB em treinamento e teste em nível de paciente, não havendo sobreposição de dados. O teste foi composto por 80 vídeos referentes a 11 pacientes totalizando 10.709 *frames*. Todos os *frames* referentes aos vídeos remanescentes foram adicionados ao conjunto de treinamento. Além disso, foi utilizada a técnica de *data augmentation* baseada no trabalho apresentado na Seção 3.1.

As redes que integram a Reg-STN apresentada pelos autores foram baseadas na arquitetura da CNN proposta no trabalho citado anteriormente. Nesse sentido, para a STN foram realizadas algumas modificações na arquitetura original, onde foi removida a camada de GAP e a camada de saída. No lugar dessas camadas foram adicionadas duas camadas totalmente conectadas para prever os parâmetros da transformação afim. A arquitetura da CNN foi deixada intacta. Ambas as redes (CNN e STN) foram treinadas por 120 épocas utilizando um tamanho do lote de 64, otimizadas com o otimizador Adam configurado com uma taxa de aprendizado de 1×10^{-4} e os mesmos parâmetros sugeridos no trabalho.

O treinamento da classificação em nível de vídeo foi realizado utilizando a técnica de validação cruzada com 5 partições. O particionamento dos dados foi realizado ao nível do paciente, não havendo sobreposição dos dados nas partições de treinamento e teste. A STN foi treinada utilizando o otimizador Adam com uma taxa de aprendizado de 1×10^{-2} e a função de perda SORD, apresentada anteriormente para classificação em nível de *frame*. O treinamento ocorreu por no máximo 30 épocas, onde foi definida uma parada precoce com base no valor do erro. Apesar de toda arquitetura poder ser treinada de ponta-a-ponta, os autores treinaram somente a camada de agregação e usaram os pesos pré-treinados da arquitetura baseada em *frames*.

A classificação em nível de *frame*, com a CNN + Reg-STN + SORD, obteve um F1-Score de 65,1%. Esse resultado indicou uma melhoria em relação a outros tipo de redes testadas no trabalho, onde o F1-Score foi de: 61,6% (CNN + Entropia Cruzada), 63,2% (CNN + SORD), 62,2% (Resnet-18 + SORD), 61% (CNN + STN + SORD), 61,8% (CNN + Random Crop + SORD).

Para a classificação em nível de vídeo utilizando uninormas, a pontuação F1-*Score* foi de 61%. Os resultados indicaram que o uso das uninormas superou outros métodos, como o máximo e a média dos valores das classificações dos *frames*, onde os valores do F1-*Score* foram de: 46% (máximo) e 51% (média).

3.2.1.2 Dastider, Sadik e Fattah (2021)

Em Dastider, Sadik e Fattah (2021), os autores apresentaram uma arquitetura de CNN que utiliza características extraídas por uma rede de autocodificadores (*autoencoders*), por blocos de convolução separável em profundidade (*depthwise separable convolution*) e de uma DenseNet-201 para a classificação em nível de *frame*. Para a classificação em nível de vídeo, os autores propuseram uma arquitetura híbrida combinando a CNN proposta com uma LSTM. O objetivo foi diagnosticar a gravidade do acometimento pulmonar causado pela COVID-19. Os autores utilizaram as mesmas pontuações (0–3) apresentadas no

trabalho da Seção 3.2.1.1.

O conjunto de dados utilizado nesse trabalho foi uma versão reduzida do ICLUS-DB (ROY et al., 2020). Ao todo foram considerados 60 vídeos referentes a 29 pacientes. Para a aquisição dos vídeos foram usados transdutores convexos e lineares. No entanto, apenas 58 dos 60 vídeos disponíveis possuíam anotações em nível de *frame*, sendo 38 de transdutores convexos e 20 de transdutores lineares. Cada *frame* foi associado a um diagnóstico de gravidade (0–3). O processo de anotação dos dados foi realizado conforme descrito no trabalho mencionado anteriormente.

Para a classificação em nível de *frame*, a rede de autocodificadores foi responsável pela redução de ruído e extração das características capazes de discriminar as quatro pontuações. Os blocos de convolução separáveis de profundidade foram responsáveis pela extração das características de baixo nível (filtro de Sobel). Essas características foram passadas ao longo da CNN para as camadas convolucionais. Por fim, as características extraídas pela base da CNN foram concatenadas à saída de uma DensetNet-201 modificada, servindo de entrada para as camadas totalmente conectadas, responsáveis pela classificação. Foram utilizadas três camadas totalmente conectadas com 128, 64 e 4 neurônios, esta última configurada com uma função de ativação *softmax*.

Para a classificação em nível de vídeo, os autores propuseram uma arquitetura híbrida composta pela CNN utilizada na classificação em nível de *frame* (autocodificadores + convolução separável de profundidade + DenseNet-201) e uma LSTM. A classificação referente a cada *frame* foi utilizada de entrada para uma LSTM e a camada de saída foi configurada com uma camada totalmente conectada com 4 neurônios e função de ativação *softmax*.

Foi utilizada a técnica de validação cruzada com cinco partições (80% para treinamento e 20% para testes). Além disso, foi proposto o uso da técnica de *data augmentation* com o seguinte pré-processamento: rotação, translação, escala, inversão horizontal e vertical. O treinamento foi realizado por 120 épocas, utilizado um tamanho do lote igual a 64 e o otimizador Adam configurado com uma taxa de aprendizado de 1×10^{-3} para ambas as redes (CNN e LSTM).

O desempenho da arquitetura proposta foi avaliado com base na acurácia, sensitividade, especificidade e F1-*Score*. Os seguintes resultados foram reportados para a classificação em nível *frame* para os transdutores lineares: 70% (acurácia), 70% (sensibilidade), 90,8% (especificidade) e 70,2% (F1-*Score*). Em relação aos transdutores convexos os resultados foram inferiores: 61% (acurácia), 61% (sensibilidade), 75,6% (especificidade) e
58,6,6% (F1-Score).

Os seguintes resultados foram reportados para a classificação em nível vídeo para os transdutores lineares: 79,1% (acurácia), 79,1% (sensibilidade), 90,1% (especificidade) e 78,6% (F1-*Score*). Em relação aos transdutores convexos os resultados também foram inferiores: 67,7% (acurácia), 67,7% (sensibilidade), 76,8% (especificidade) e 66,6% (F1-*Score*).

A arquitetura híbrida (CNN-LSTM) obteve um resultado superior ao apresentado pela CNN proposta, onde o incremento na acurácia foi de 7–9%. A classificação em nível de vídeo pela LSTM forneceu melhores resultados no geral.

3.2.2 COVID-19 Lung Ultrasound Dataset

Os trabalhos organizados nesta seção utilizaram o mesmo conjunto de dados, o COVID-19 Lung Ultrasound Dataset (BORN; BRÄNDLE et al., 2020), disponível em https:// github.com/jannisborn/covid19_ultrasound (acessado em 18 de dezembro de 2021). No entanto, esse conjunto de dados é constantemente atualizado, portanto, podem existir diferenças significativas na quantidade de dados utilizados em cada trabalho.

3.2.2.1 Horry et al. (2020)

Em Horry et al. (2020), os autores realizaram um estudo com diferentes tipos de imagens, dentre elas imagens de USP, para propor uma arquitetura multimodal de aprendizado profundo para classificação em nível de *frame*. Os autores apresentaram duas propostas de classificação: uma, considerando a junção das classes COVID-19 e pneumonia bacteriana versus a classe saudável, e a outra, considerando isoladamente a classe COVID-19 versus a classe pneumonia bacteriana sem a presença dos casos saudáveis.

Foi utilizado o conjunto de dados COVID-19 *Lung Ultrasound Dataset*. Ao todo foram extraídos dos vídeos 911 *frames*. No geral, os *frames* apresentaram uma grande variação na qualidade, no tamanho, no contraste e no brilho. Para evitar o viés provocado pela diferença na intensidade dos *pixels* dos *frames*, foi aplicado um método conhecido como N-CLAHE (do inglês *Contraste Limited Adaptive Histogram Equalization*) (KOONSANIT et al., 2017) para realçar pequenos detalhes, texturas e contraste local.

Os *frames* foram redimensionados para diferentes resoluções para fins de compatibilidade com as arquiteturas pré-treinadas utilizadas. Para aumentar o número de *frames*, foi aplicada a técnica de *data augmentation* com o seguinte pré-processamento: inversão horizontal e vertical, rotação, e translação vertical e horizontal.

O conjunto de dados foi divido em 80% para o treinamento e 20% para testes. Foi utilizada a técnica de transferência de aprendizado, compensando a quantidade de dados e diminuindo o tempo de treinamento. Os hiperparâmetros foram otimizados com base em diferentes arquiteturas de CNNs fornecidas pela biblioteca *Keras*: VGG-16/19, ResNet50V2, InceptionV3, Xception, InceptionResNetV2, NASNetLarge e DenseNet121. Para a otimização dos hiperparâmetros foi utilizada a biblioteca *Keras-Tuner*, onde os seguintes intervalos de hiperparâmetros foram testados: taxa de aprendizado $(1 \times 10^{-5} \text{ a} 1 \times 10^{-3})$, tamanho da camada oculta (8 a 96 neurônios), taxa de *dropout* (0,1 a 0,2) e tamanho do lote (2 a 16). O treinamento foi inicialmente executado por 100 épocas.

Após a otimização dos hiperparâmetros e a comparação das arquiteturas, os autores verificaram que a VGG19 obteve melhor desempenho no geral. Dessa forma, foi realizada uma nova bateria de testes alterando o intervalo dos seguintes hiperparâmetros: taxa de aprendizado $(1 \times 10^{-6} \text{ a } 1 \times 10^{-3})$ e o tamanho da camada oculta (4 a 96 neurônios). Os resultados demonstraram que o *droupot* teve pouca influência na acurácia das redes, exceto quando utilizadas taxas de aprendizado muito baixas (1×10^{-6}) ou altas (1×10^{-3}) , onde a taxa de *dropout* de 0,2 provou-se mais estável. Já os hiperparâmetros taxa de aprendizado, tamanho do lote e quantidade de neurônios na camada totalmente conectada afetaram a acurácia da rede.

A VGG19 que foi treinada em *frames* com base na junção das classes COVID-19 e pneumonia bacteriana versus a classe saudável obteve uma sensibilidade de 97%, precisão de 99% e F1-*Score* de 98%. A VGG19 que considerou os *frames* isolados da classe COVID-19 versus pneumonia bacteriana, obteve uma sensibilidade, precisão e F1-*Score* de 100%. Para ambas as arquiteturas, a configuração final utilizada foi uma taxa de aprendizado de 1×10^{-5} , taxa de *dropout* de 0,2, tamanho do lote de 2, e 64 neurônios na camada totalmente conectada.

A VGG19 selecionada apresentou resultados consideráveis, mostrando uma boa capacidade de distinção entre as classes de diagnóstico.

3.2.2.2 Born, Wiedemann et al. (2021)

No trabalho apresentado em Born, Wiedemann et al. (2021), os autores propuseram a criação de um conjunto de dados com base em dados disponíveis publicamente. Além da criação desse conjunto de dados, os autores sugeriram uma CNN em nível de *frame* e

vídeo para a classificação de três classes de doenças pulmonares: COVID-19, pneumonia bacteriana e saudável.

O conjunto de dados foi criado a partir de hospitais, publicações científicas, plataformas comunitárias, repositórios médicos e empresas médicas. Foram coletadas imagens de transdutores lineares e convexos. Os autores publicaram a primeira versão desse conjunto de dados em Born, Brändle et al. (2020). Ao todo foram coletados 202 vídeos e 59 imagens (algumas delas retiradas de publicações científicas) referentes a 216 pacientes de 41 fontes diferentes. Desses 202 vídeos, 101 foram capturados utilizando o protocolo BLUE. Além das três classes mencionadas (COVID-19, pneumonia bacteriana e saudável), foram coletados mais 6 vídeos da classe pneumonia viral.

Os autores optaram por não utilizar os vídeos da classe pneumonia viral, pois o número de vídeos não era expressivo para ser considerado. Além disso, os vídeos adquiridos por transdutores lineares também não foram usados, optaram pelos vídeos adquiridos por transdutores convexos, pois segundo os autores são mais indicados para o US à beira do leito. Dessa forma, foram utilizados 179 vídeos no experimento e 53 imagens. Os vídeos e *frames* foram revisados e aprovados por dois médicos especialistas (um médico pediatra com mais de 10 anos de experiência em USP e um instrutor acadêmico de US).

Os *frames* referentes aos 179 vídeos foram extraídos a uma taxa de 3 Hz (máximo de 30 frames por vídeo), gerando 1.204 frames de COVID-19, 704 frames de pneumonia bacteriana e 1.326 *frames* da classe saudável. Além da extração, os *frames* foram redimensionados para 224×224 *pixels* e as informações desnecessárias como barras de medida, textos e logos de empresas foram removidas.

A classificação em nível de *frame* foi realizada por uma VGG-16 e uma NasNetMobile, esta última considerada uma alternativa leve, requerendo menos parâmetros do que a VGG-16, onde o foco foi nos dispositivos portáteis. Além disso, foram utilizadas duas variações da VGG-16 que os autores chamaram de VGG e VGG-CAM. A VGG-CAM foi composta de uma única camada densa após a camada de GAP, permitindo assim o uso de mapas de ativação de classe simples (CAMs), enquanto a VGG (mesma arquitetura da POCOVID-Net) foi composta por uma camada densa adicional (64 neurônios) com função de ativação ReLU, uma camada de *dropout* com uma taxa de 0,5, uma camada de (*batch normalization*) e uma camada de saída configurada com a função softmax.

Foi utilizada a técnica de validação cruzada com 5 partições e os dados estratificados em nível de paciente, evitando sobreposição de dados do mesmo paciente nas partições de treinamento e teste. Além disso, o número de vídeos por classe se manteve proporcional nas partições. No treinamento também foi utilizada a técnica de *data augmentation* com o seguinte pré-processamento: inversão, rotação e translação.

Foi realizado um ajuste fino somente nas últimas três camadas de ambas as redes, onde foram mantidos os pesos do pré-treinamento para as demais camadas. O resultado foi uma quantidade de $\approx 2,4$ M de parâmetros treináveis e $\approx 12,4$ M de parâmetros não treináveis.

As redes foram treinadas por 40 épocas, com tamanho do lote igual a 8, função de perda de entropia cruzada e uma parada precoce com base no erro para evitar o sobreajuste. Foi utilizado o otimizador Adam configurado com uma taxa de aprendizado de 1×10^{-4} .

A classificação em nível de vídeo utilizou as mesmas redes propostas em nível de *frame*, porém a classificação final foi dada pela médias dos valores das classificações obtidas em cada *frame*. Os autores também propuseram comparar com outra rede conhecida como Models Genesis (ZHOU, Z. et al., 2019) (uma arquitetura para análise de imagens 3D pré-treinadas em TC de pulmão), onde os vídeos foram divididos em blocos de 5 *frames*.

Os resultados das classificações em nível de *frame* mostraram que entre as arquiteturas experimentadas (VGG, VGG-CAM e NASNetMobile), a VGG atingiu a melhor acurácia (87,8%), seguida pela VGG-CAM (87,4%) e NASNetMobile (62.5%). A VGG também obteve os melhores resultados em relação às medidas de sensibilidade (88%), precisão (90%), F1-*Score* (89%) e especificidade (94%) para a COVID-19. Apesar da NASNetMobile necessitar de bem menos parâmetros do que a VGG ($\approx 1/3$), a acurácia e demais medidas foram baixas, mostrando que esse tipo de arquitetura não produziu bons resultados.

Os resultados das classificações em nível de vídeo indicaram que a arquitetura Models Genesis (CNN 3D) treinada em imagens de TC de pulmão, obteve um desempenho inferior (78% de acurácia) ao da VGG utilizando a média dos valores das saídas referentes a cada frame (90% de acurácia). Os resultados referentes à VGG para a COVID-19 foram: 90% (acurácia), 92% (precisão), 91% (F1-score) e 96% (especificidade).

3.2.2.3 Awasthi et al. (2021)

Em Awasthi et al. (2021), os autores apresentaram uma CNN com uma arquitetura eficiente e leve (em termos de parâmetros e memória) baseada nas MobileNets, chamada Mini-COVIDNet. O foco do trabalho foi apresentar uma rede que fosse capaz de rodar em aplicativos móveis ou embarcados para a classificação em nível de *frame* de três classes de doenças pulmonares: COVID-19, pneumonia bacteriana e saudável.

O conjunto de dados utilizado no trabalho foi o COVID-19 *Lung Ultrasound Dataset* e consistiu de 64 vídeos, sendo 11 destes pertencentes a pacientes saudáveis, 14 a pacientes com pneumonia bacteriana e 39 com COVID-19. Foram extraídos 182 *frames* pertencentes à classe saudável, 277 à classe pneumonia e 678 à classe COVID-19.

Para combater o sobreajuste devido ao tamanho do conjunto de dados, os autores propuseram o uso da técnica de validação cruzada com cinco partições. A distribuição do número de *frames* por classe foi mantida entre as diferentes partições. Além disso, a divisão dos dados foi feita em nível de paciente. No treinamento foi utilizada a técnica de *data augmentation* com o seguinte pré-processamento: rotação, inversão horizontal e vertical, e escala.

Neste trabalho os autores propuseram uma modificação na MobileNet, onde combinaram essas modificações com uma função de perda focal (*focal loss*). Essa função teve como objetivo combater o desiquilíbrio no treinamento entre os exemplos das classes. A perda focal adiciona um termo a função de perda de entropia cruzada. Dessa forma, a função de entropia cruzada é escalonada dinamicamente, tal que esse fator de escala decai a zero à medida que a confiança na classe correta aumenta. Na prática, esse fator de escala pode diminuir automaticamente a contribuição de exemplos fáceis durante o treinamento e rapidamente focar em exemplos difíceis, conforme (LIN et al., 2017).

A MobileNet foi treinada no conjunto de dados ImageNet e os pesos foram mantidos. No entanto, as últimas três camadas sofreram um ajuste fino nos pesos. Uma camada completamente conectada com 64 neurônios e função de ativação ReLU foi adiciona a rede pré-treinada. Foi configurada uma camada de *dropout* com uma taxa de 0,5, seguida por uma camada de *batch normalization*. Na saída foi utilizada uma camada completamente conectada contendo três neurônios e a função *softmax*.

O treinamento foi realizado por 50 épocas, utilizando o otimizador Adam configurado com uma taxa de aprendizado de 1×10^{-4} para fins de comparação. Foi utilizado o *framework Tensorflow* e *Tensorflow Lite*, este último com foco em dispositivos móveis, internet das coisas e sistemas embarcados. Além da rede proposta, foram testadas outras arquiteturas como: COVID-CAPS, COVID-CAPS Scaled, POCOVID-Net, MobileNet, NasNetMobile e ResNet50. Para cada uma dessas arquiteturas foram testadas as configurações com e sem a função de perda focal. O processamento foi realizado em uma estação de trabalho Linux, que consistiu em uma CPU Intel Xeon Silver 4110 com velocidade de clock de 2,10 GHz, 128 GB de RAM e uma GPU NVIDIA Titan RTX com 24 GB de memória. Adicionalmente, para os testes em dispositivos integrados, foram utilizados dois tipos de dispositivos de baixo custo. Um Raspberry Pi 4 Modelo B e um NVIDIA Jetson AGX Xavier.

Os autores demonstraram que o uso da função de perda focal na arquitetura proposta (Mini-COVIDNet) quando comparada com a sua mesma versão utilizando a função de entropia cruzada, ajudou a melhorar os resultados. No entanto, esse comportamento não pode ser generalizado para as outras arquiteturas testadas, pois algumas apresentaram perda de acurácia quando a função foi utilizada. Porém, para o caso da arquitetura apresentada foi registrado um incremento nas medidas de avaliação.

A quantidade de memória requerida pela rede COVID-CAPS foi muito menor se comparada com as outras redes, mas o desempenho em relação à precisão e F1-*Score* foi pior. O desempenho da rede POCOVID-Net foi similar ao Mini-COVIDNet, enquanto o número de parâmetros na arquitetura Mini-COVIDNet foi 4,39 menor (3,3 M de parâmetros), resultado em uma rede de tamanho menor (51,29 MB).

Em relação aos resultados do desempenho nos dispositivos integrados, as arquiteturas Mini-COVIDNet e MobileNet apresentaram as latências mais baixas entre todas as arquiteturas testadas. A latência para o dispositivo Raspberry Pi 4 Modelo B apresentada pela rede MobileNetV2 utilizando o *framework TensorFlow Lite* foi de 0,1465 segundos. A mesma arquitetura também apresentou os melhores resultados para o dispositivo NVI-DIA Jetson AGX Xavier, cujo o resultado foi de 0,066 segundos de latência para o mesmo *framework*.

O uso do *framework Tensorflow* (versão padrão) resultou em um incremento de latência 7 vezes maior do que o resultado obtido pela versão *lite* do *framework*, para todas as arquiteturas. Além dos resultados referentes à latência, também foram reportados dados sobre o consumo de memória das redes nesses dispositivos. A rede com menor consumo de memória foi a COVID-CAPS (1 MB), seguida pela MobileNetV2 (9 MB) e a Mini-COVIDNet (13 MB).

Quanto ao tempo de treinamento, tanto as MobileNets quanto as Mini-COVIDNet tiveram bons resultados e foram pelo menos 1,7 vezes mais rápidas do que as outras arquiteturas. Conforme relatado pelos autores, tanto as MobileNets quanto as Mini-COVIDNet apresentaram bons resultados, pois foram mais rápidas pelo menos 1,7 vezes do que as demais. A arquitetura mais rápida foi a MobileNetV2 (com função de entropia cruzada) levando 23,57 minutos em um dispositivo NVIDIA Jetson AGX Xavier. A arquitetura Mini-COVIDNet obteve um resultado similar de 26,20 minutos.

No geral, a Mini-COVIDNet (com função de perda focal) apresentou um bom custo benefício para os dispositivos integrados, pois obteve resultados expressivos em relação ao seu tamanho, latência, tempo de treinamento e acurácia (utilizada para a comparação). A acurária da Mini-COVIDNet foi de 83,2%, superando as versões da MobileNetV2 (79,3% sem função de perda focal e 76,3% com perda focal) e se igualando a POCOVID-Net (83,2% com função de perda focal), ambas mostrando um empate em relação à acurácia, o que faz da rede Mini-COVIDNet uma arquitetura promissora para uso em dispositivos mais leves.

3.2.2.4 Muhammad e Hossain (2021)

Em Muhammad e Hossain (2021), os autores sugeriram a utilização de uma CNN com fusão de características multicamada para a classificação em nível de *frame* de três classes de doenças pulmonares: COVID-19, pneumonia bacteriana e saudável.

O trabalho usou o conjunto de dados COVID-19 *Lung Ultrasound Dataset* contendo imagens adquiridas por transdutores convexos e lineares. Entretanto, para o experimento foram considerados somente os dados adquiridos por transdutores convexos (maior quantidade) sendo desconsiderados os dados da empresa Butterfly. Ao todo foram utilizados 121 vídeos.

A arquitetura da CNN proposta foi composta por um módulo chamado ResF, posicionado no início da rede e cinco blocos contendo dois módulos ResF e um conector de módulo ResF referente à conexão de salto (*skip connection*). O módulo ResF foi composto por uma camada convolucional com *batch normalization* e função de ativação ReLU. O módulo ResF posicionado no início da rede foi configurado com filtros 5×5 e os blocos 1 a 5 com filtros 3×3 , com exceção dos módulos ResF usados nas conexões de salto (1×1) a cada bloco. Após o conector ResF foi utilizada uma camada de *average pooling* e após a saída do quinto bloco, foi utilizada uma camada de GAP, servindo de entrada para as camadas completamente conectadas, sendo a última configurada com a função *softmax* para classificação.

Existem três níveis de fusão que podem ser utilizados para melhoria da acurácia. O primeiro considera o fluxo de dados de múltiplas fontes, onde essas características referentes a cada um dos fluxos são concatenadas para produzir um vetor de características submetido a um classificador. O segundo nível refere-se à fusão em nível de classificação, onde as características de cada fluxo são alimentadas por classificadores diferentes, cujas saídas dos classificadores são fundidas. O terceiro nível usa a fusão com base na decisão, onde cada classificador produz uma decisão que serve como base para uma votação final.

O problema da fusão de dados de múltiplas fontes é que o número de parâmetros acaba aumentando demais e consequentemente a rede passa a necessitar de mais recursos computacionais e tempo de treinamento. De modo a evitar esse problema, os autores propuseram uma fusão com base nas saídas dos blocos convolucionais 1 a 5, onde essas saídas foram fundidas e serviram como entrada para a camada de classificação, evitando assim o aumento drástico da quantidade de parâmetros treináveis. Segundo os autores, o objetivo central da utilização da técnica de fusão multicamadas, está no fato de que cada camada extrai um tipo diferente de características dessas imagens, portanto, a fusão dessas características pode aumentar a acurácia do classificador.

No pré-processamento, as imagens foram redimensionadas para 512×512 e os *pixels* foram padronizados, onde cada valor de *pixel* foi subtraído da média dos valores e depois divididos pelo desvio padrão. Além da padronização dos *pixels*, foi utilizada a técnica de *data augmentation* com o seguinte pré-processamento: inversão horizontal e vertical, rotação e escala. Segundo os autores, essas foram as transformações naturais aos exames de USP.

No treinamento foi utilizada a técnica de validação cruzada com cinco partições. Foi utilizado um tamanho do lote de 5 e o otimizador Adam configurado com uma taxa de aprendizado de 5×10^{-4} , beta_1 de 0,9, beta_2 de 0,999 e epsilon de 1×10^{-7} . Essas configurações foram as melhores dentre os parâmetros testados. Além disso, a quantidade de camadas completamente conectadas e a quantidade de neurônios nessas camadas também foram variadas.

Os resultados foram avaliados em termos da acurácia, precisão, sensibilidade e AUC (do inglês *Area Under ROC Curve*). Foram comparadas as arquiteturas pré-treinadas ResNet50 e SqueezeNet, e a arquitetura proposta de CNN com fusão e sem fusão. Os resultados indicaram que a arquitetura proposta com fusão superou todas as outras, mostrando um incremento em todas as medidas avaliadas. A acurácia reportada foi de 92,5%, precisão de 91,8%, sensibilidade de 93,2% e AUC de 99,93%. O número de parâmetros treináveis na arquitetura proposta com e sem fusão foram bem próximos 0,42M e 0,40M, mostrando pouca diferença entre elas. Entretanto, em relação às outras arquiteturas pré-treinadas a diferença foi significativa: 1,24M (SqueezeNet) e 25,6M (ResNet50).

3.2.3 Outros Conjuntos de Dados

Os trabalhos organizados nesta seção utilizaram diferentes conjuntos de dados, porém não foram disponibilizados publicamente.

3.2.3.1 Jiaqi Zhang et al. (2020)

No trabalho apresentado em Jiaqi Zhang et al. (2020) foi proposto o uso de três tipos de arquiteturas de CNNs para a classificação em nível de *frame* de diferentes achados radiológicos de pneumonia, com o objetivo de auxiliar no diagnóstico da COVID-19.

O conjunto de dados foi composto por 10.350 imagens. As imagens foram colectadas de diferentes fontes. Cada imagem passou por uma etapa de reclassificação e de reanotação, onde cada uma delas foram verificadas e seus devidos rótulos associados a 8 tipos de achados radiológicos (0–7), são eles: normal (0), quantidade de linhas B inferior a 3 (1), quantidade de linhas B superior a 3 (2), área de fusão da linha B é inferior à metade (3), área de fusão da linha B é superior à metade (4), profundidade das peças é inferior a 1 cm (5), broncograma aéreo e a profundidade de hepatização são inferiores a 3 cm (6) e derrame pleural e profundidade de hepatização é superior a 3 cm (7).

Como cada estágio da pneumonia foi caracterizado por mais de um desses achados, os autores sugeriram a formação de três grupos, cada um contendo uma quantidade de características para representar um estágio clínico da pneumonia. O primeiro grupo foi composto por 3 classes (0, 1-4 e 5-7), o segundo grupo por 4 classes (0, 1-4, 5-6 e 7) e o terceiro pelas 8 classes (0-7). Foi adotada uma proporção de 90% das imagens para treinamento e 10% para testes.

As arquiteturas de CNNs utilizadas no treinamento foram: VGG-19, ResNet-101 e EfficientNet-B5. No entanto, para acelerar o treinamento foi utilizada a técnica de transferência de aprendizado. As arquiteturas VGG-19 e ResNet-101 foram pré-treinadas no conjunto de dados ImageNet e CIFAR-100 (KRIZHEVSKY; HINTON, 2009). A arquitetura EfficientNet-B5 foi treinada em oito grandes conjuntos de dados e 1 conjunto de dados privado de ultrassom de câncer de mama (não foram fornecidas maiores explicações sobre os conjuntos de dados).

As redes pré-treinadas foram utilizadas como extratores de características, onde a camada de classificação foi substituída para representar a nova quantidade de classes. Todas as redes sofreram um ajuste fino. Para reduzir o desbalanceamento entre as classes, foi utilizada a técnica de *data augmentation* com o seguinte pré-processamento: corte aleatório e inversão horizontal. Os seguintes hiperparâmetros foram otimizados: taxa de aprendizado $(2 \times 10^{-4} \text{ a } 1 \times 10^{-2})$, tamanho do lote (3 a 24) e os otimizadores (SGD e Adam) foram explorados buscando a melhor acurácia. Para todas as redes, a função de perda utilizada foi a de entropia cruzada, com exceção da EfficientNet-B5 (8 classes ou grupo 3) que utilizou a função de perda de entropia cruzada ponderada.

As redes que utilizaram a EfficientNet-B5 como base (3, 4 e 8 classes) convergiram com 50 épocas e levaram um tempo de treinamento máximo de 2,5 horas. Após as rodadas de otimização, as seguintes configurações foram adotadas: taxa de aprendizado de 2×10^{-4} , tamanho do lote de 16 e otimizador Adam.

As VGGs-19 e ResNet-101 convergiram com 250 épocas e levaram respectivamente 3 horas para os grupos contendo 3 e 4 classes e 8 horas para o grupo contendo 8 classes. Em relação aos hiperparâmetros, foram adotadas as seguintes configurações: taxa de aprendizado de 1×10^{-4} para ambas as arquiteturas, tamanho do lote de 12 para as VGGs e 24 para as ResNets, otimizador SGD com momentum de 0,9 para as VGGs e Adam para as ResNets.

Para avaliar as diferentes arquiteturas, os autores compararam a acurácia. Segundo esse critério, as EfficientNets obtiveram melhor desempenho. A acurácia para 3, 4 e 8 classes foi de 94,6%, 91,2% e 82,3%. As VGGs apresentaram os seguintes resultados: 89,1%, 88,4% e 60%, respectivamente. As ResNets apresentaram os resultados de 88,6%, 87,5% e 62,4%, superando as VGGs apenas para as 8 classes.

Além desse comparativo entre as redes, os autores apresentaram detalhes sobre o treinamento final das EfficientNets, onde o otimizador Adam foi configurado com um **step-size** igual a 7 e um **gamma** de 0,1. Para combater o desbalanceamento das classes, foi reportada uma modificação na função de perda. Para a classificação das 8 classes, foi utilizada a função de perda de entropia cruzada ponderada e o treinamento foi realizado utilizando validação cruzada com dez partições.

Os resultados das EfficientNets (3, 4 e 8 classes) foram: 93,2%, 89,9% e 81,6% (F1-Score); 95,5%, 95,0% e 95,4% (acurácia); 93,2%, 89,9% e 81,6% (sensibilidade); 96,6%, 96,6% e 97,4% (especificidade); e 93,2%, 89,9% e 81,6% (precisão).

As EfficientNets alcançaram um desempenho equiparável ao de especialistas humanos, com potencial para aliviar a carga de trabalho dos médicos e permitir identificar rapidamente pacientes com pneumonia.

3.2.3.2 Tsai et al. (2021)

Em Tsai et al. (2021) foi proposto o uso de uma CNN + Reg-STN para detecção do derrame pleural com foco na COVID-19. Para isso, foi utilizado um treinamento supervisionado e outro fracamente supervisionado em nível de *frame* e vídeo. No entanto, os autores mencionaram que embora não seja suficiente para o diagnóstico da COVID-19, o derrame pleural é potencialmente associado à COVID-19.

A partir dos dados coletados no *Royal Melbourne Hospital*, onde foram adquiridos 623 vídeos utilizado um protocolo padronizado baseado em seis diferentes zonas anatômicas, foram gerados um mínimo de 6 vídeos por paciente, um para cada uma dessas regiões anatômicas, totalizando 99.209 *frames*.

O pulmão de um paciente foi considerado normal se nenhum dos vídeos referentes às zonas anatômicas mostraram sinais de derrame pleural. No entanto, se pelo menos uma das zonas apresentou sinais de derrame pleural, esse pulmão foi categorizado como anormal. Dos 99.209 *frames*, 20.120 (20%) apresentaram sinais de derrame pleural e 70.089 foram considerados normal (80%).

Esses vídeos foram adquiridos usando um sistema de aquisição Sonosite X-Porte ultrasound (Fujifilm, Bothell, WA, USA) e um transdutor do tipo phased array (SonoSite X-Porte rP19xp) com uma frequência de 1–5 MHz. Segundo os autores, o uso deste tipo de transdutor se deu pelo fato de alcançar maior profundidade de penetração, permitindo a visualização de grandes derrames e consequentemente fornecendo uma melhor capacidade de quantificação desse volume para o algoritmo. Os vídeos foram armazenados utilizando o formato de arquivamento conhecido como DICOM (do inglês, Digital Imaging and Communications in Medicine).

No pré-processamento, os dados no formato DICOM foram extraídos e os *frames* convertidos do espaço de cores YBR_FULL_422 para o espaço de cores RGB. Uma variedade de informações como texto, marcas d'água e marcas registradas do sistema de imagem foram substituídas por *pixels* na cor preta. As imagens foram cortadas em um formato mínimo retangular, contendo somente os dados de interesse, reduzindo o tamanho dos dados de entrada.

As anotações foram feitas por um sistema automatizado (iLungScan^M) desenvolvido pelo grupo de educação em US da Universidade de Melbourne. O sistema foi validado por especialistas e clínicos e consegue indicar sinais de derrame pleural, tendo sido validado nas seis zonas anatômicas. Cada vídeo e cada *frame* foi anotado com a respectiva classe (presença ou ausência de derrame pleural).

O conjunto de dados foi divido em conjunto de treinamento contendo 90% dos dados (63 pacientes) e um conjunto de teste contendo os 10% restantes dos dados (7 pacientes). Foi utilizada a técnica de validação cruzada com dez partições, onde cada paciente apareceu pelo menos uma vez no conjunto de testes. A distribuição das classes normal e anormal no conjunto de treinamento para os vídeos foi de 80% e 20% e para os *frames* foi de 84% e 16%.

Os autores propuseram a utilização da arquitetura CNN + Reg-STN, baseada no trabalho apresentado na Seção 3.2.1.1. Essa arquitetura tem como objetivo diagnosticar a gravidade do acometimento pulmonar causado pela COVID-19 (com base numa pontuação que varia de 0–3). No entanto, para que fosse possível atender ao objetivo proposto pelos autores, foi realizada uma modificação na arquitetura original, adequando-a à classificação binária (presença ou ausência do derrame pleural).

A CNN + Reg-STN foi treinada por 120 épocas conforme o proposto na Seção 3.2.1.1, sendo utilizado o otimizador Adam e um tamanho do lote igual a 64. Os resultados foram avaliados em termos da acurácia, F1-*Score*, precisão e sensibilidade. A acurácia média foi de 92.38% para a classificação em nível de *frame*, onde foi 1.26% maior do que a classificação em nível de videos (91,12%).

Os outros resultados foram avaliados e expressos em termos do pior e do melhor valor obtido nas partições de teste. Os valores da classificação em nível de *frame* foram: acurácia de 86,30%-96,75%, F1-*Score* de 34,98%-90.47%, precisão de 42.82%-92.76% e sensibilidade de 29,57%-88.24%. Os valores da classificação em nível de vídeo foram: acurácia de 84,58%-95,68%; F1-*Score* de 40,02%-87,71%; precisão de 38.85%-87.29%; sensibilidade de 41,26%-88.14%.

Os resultados indicaram que a CNN + Reg-STN pode ser utilizada para diagnosticar o derrame pleural com uma acurácia equiparável aos padrões clínicos, permitindo um diagnóstico mais rápido e robusto, independentemente das habilidades do operador do US.

3.2.3.3 Arntfield et al. (2021)

No trabalho apresentado em Arntfield et al. (2021) foi proposta a utilização de uma CNN Xception para classificação em nível de *frame* e vídeo de diferentes doenças que se correlacionam com as linhas B: COVID-19 com SDRA (síndrome do Desconformo Respiratório Agudo), não COVID-19 com SDRA e edema pulmonar hidrostático (EPH).

O conjunto de dados foi composto por uma variedade de aquisições oriundas de diferentes sistemas de US, sendo o transdutor *phased array* o mais utilizado neste conjunto (predominante na América do Norte). O formato utilizado para os vídeos foi o MPEG-4 (do inglês *Moving Picture Experts Group*), variando entre 3 a 6 segundos e um *frame rate* entre 30 a 60 *frames* por segundo (dependendo do sistema de US).

No conjunto de dados usado pelos autores, o número de exames com diagnóstico da COVID-19 foi menor que o das outras classes. Os casos de COVID-19 foram confirmados através do teste RT-PCR. Foram utilizados 612 vídeos, totalizando 121.381 *frames* pertencentes a 243 pacientes.

No pré-processamento dos dados, os *frames* referentes a cada vídeo foram extraídos, redimensionadas para 600×600 *pixels* e convertidos para o sistema de cores RGB. Esse redimensionamento foi necessário para manter a compatibilidade com a rede pré-treinada Xception. As informações desnecessárias, e que, poderiam prejudicar o aprendizado, como as marcas de índice, logotipos de fabricantes entre outras, foram removidas.

A técnica de *data augmentation* foi utilizada para combater o sobreajuste dos dados. Foi realizado o seguinte pré-processamento: zoom aleatório, inversão horizontal, alongamento e contração horizontal e vertical, e rotação bidirecional.

O conjunto de dados foi divido aleatoriamente em um conjunto de treinamento e dois conjuntos de testes (teste 1 e teste 2). O teste 1 foi usado para avaliar todas as redes candidatas e para a escolha dos hiperparâmetros. O teste 2 foi usado para avaliação da rede final, não existindo sobreposição de dados entre os conjuntos de treinamento, teste 1 e teste 2.

Foram realizadas pequenas modificações na arquitetura da rede Xception. A saída das camadas convolucionais serviram de entrada para uma camada de GAP, resultando em um vetor de características. Uma camada de *dropout* com uma taxa de 0,6 foi aplicada para introduzir regularização à rede e evitar o sobreajuste. A camada de classificação foi composta por uma camada totalmente conectada com três neurônios e função de ativação softmax. Foi utilizada uma parada precoce, para interromper o treinamento caso o valor do erro no conjunto de validação não diminuísse num intervalo de 15 épocas, parando o treinamento caso contrário.

O desempenho da CNN proposta foi determinada pelo conjunto de dados de teste 2. O resultado foi avaliado em nível de *frame* e vídeo, onde este último foi alcançado através da média das probabilidades previstas pelo classificador em todos os *frames* do vídeo. O desempenho da rede final foi avaliada com base na sensibilidade, especificidade, precisão, F1-*Score* e AUC.

Os seguintes resultados foram reportados para as classes (COVID-19 com SDRA, não COVID-19 com SDRA e EPH) em nível de *frame*: 92,4%, 76,0% e 69,3% (sensibilidade); 88,3%, 81,5% e 99,9% (especificidade); 71,3%, 73,1% e 99,6% (precisão); 80,5%, 74,6% e 81,7% (F1-*Score*); e 96,5%, 89,3% e 99,1% (AUC).

Os resultados em nível de vídeo para as classes (COVID-19 com SDRA, não COVID-19 com SDRA e EPH) foram: 100%, 85,7%, 57,1% (sensibilidade); 92,9%, 76,9% e 100% (especificidade); 85,7%, 66,7%, 100% (precisão); 92,3%, 75% e 72,7% (F1-*Score*) ; 100%, 93,4% e 100% (AUC).

3.3 Conclusões

Os trabalhos apresentados no contexto das doenças pulmonares e da COVID-19 são bem diversificados em relação às técnicas de aprendizado profundo. Foram utilizados diferentes tipos de arquiteturas para classificação em nível de *frame* e vídeo. Além disso, foram considerados muitos objetivos distintos, desde a detecção de uma linha B até o diagnóstico da gravidade do acometimento pulmonar causado pela COVID-19. Um resumo desses trabalhos é apresentado na Tabela 18 do Apêndice A. A Seção 3.3.1 apresenta as principais arquiteturas utilizadas nos trabalhos em nível de *frame* e a Seção 3.3.2 em nível de vídeo.

3.3.1 Classificação em Nível de Frame

Em relação à classificação em nível de *frame*, as seguintes arquiteturas foram citadas: DenseNet-121, Densenet-201, EfficientNet-B5, Inception V2, Inception V3, InceptionRes-NetV2, NASNetLarge, NASNetMobile, POCOVID-Net, ResNet50V2, ResNet-101, SqueezeNet, VGG16/19 e Xception. Além dessas arquiteturas, foram utilizadas as SSDs, STNs, CNNs com blocos autocodificadores (*autoencoders*), com blocos de convolução separável em profundidade (*depthwise separable convolution*) e CNNs com fusão multicamada. Também foram utilizadas diferentes funções de perda, como a função de perda focal e SORD. No geral, as redes apresentadas forneceram bons resultados para diferentes achados radiológicos, tais como: linhas A, linhas B, linhas B coalescentes, consolidações, derrame pleural, deslizamento pleural, linha pleural irregular, entre outros.

3.3.2 Classificação em Nível de Vídeo

Para a classificação em nível de vídeo, alguns trabalhos utilizaram variações das arquiteturas em nível de *frame*, considerando a média das classificações obtidas em cada um dos *frames* para compor a predição final. Esse foi o caso dos trabalhos apresentados na Seção 3.2.2.2 e 3.2.3.3. Uma variação dessa abordagem foi apresentada na Seção 3.2.1.1, onde o trabalho utilizou uma camada uninormas capaz de receber as classificações realizadas em diferentes *frames*, agregando-as em uma única saída, flexibilizando o uso da média das classificações como uma solução para a classificação em nível de vídeo.

Na Seção 3.1.3, o trabalho apresentado considera o uso de uma CNN 3D que utiliza 12 frames como entrada para a classificação da presença de linhas B e da gravidade do acometimento pulmonar com base nessas linhas. Na Seção 3.2.2.2, utilizaram uma CNN 3D chamada *Models Genesis*, pré-treinada em TC de pulmão utilizando cinco frames como entrada para diagnosticar a COVID-19 e a pneumonia bacteriana. Apesar da Seção 3.1.3 apresentar uma CNN 3D com bom desempenho, esse resultado não foi repetido com a Models Genesis, conforme descrito na Seção 3.2.2.2. Uma das possíveis causas apresentadas pelos autores é que as CNNs 3D são normalmente treinadas em volumes, como as imagens de TC. Dessa forma, elas podem não ser tão propícias para tratar sequências de exames 2D de US.

Na Seção 3.2.1.2 foi apresentada uma arquitetura híbrida (CNN-LSTM), onde cada frame de um vídeo foi submetido a uma CNN (autocodificadores + convolução separável de profundidade + DenseNet-201) e a saída contendo a classificação do frame foi agregada (302 frames), servindo de entrada para uma LSTM realizar o diagnóstico da gravidade do acometimento pulmonar causado pela COVID-19. No entanto, a LSTM não utilizou características espaciais como entrada, apenas as classificações em nível de frame realizadas pela CNN.

3.4 Contribuições do Trabalho

O método proposto nesta dissertação se diferencia dos demais por considerar as características espaciais e temporais presentes nas sequências de *frames* dos vídeos de USP. De maneira análoga à arquitetura híbrida (CNN-LSTM) apresentada na Seção 3.2.1.2, que utiliza as classificações dos *frames* como entrada para uma LSTM, este trabalho propõem extrair características espaciais através das camadas convolucionais de uma CNN. Dessa forma, essas características podem ser utilizadas como entrada para uma LSTM, capaz de aprender as características temporais. Além disso, propõe-se testar as principais arquiteturas de CNNs pré-treinadas no conjunto de dados ImageNet e USP disponíveis, com o intuito de selecionar o melhor extrator de características. Por fim, propõe-se investigar o impacto da quantidade de *frames* extraídos dos vídeos nos resultados do método proposto.

4 Método para Classificação de Vídeos de Ultrassom Pulmonar

Este capítulo apresenta o método proposto para classificação de vídeos de USP. Na Seção 4.1 são apresentados os detalhes sobre a construção do conjunto de dados utilizado nesta dissertação. A Seção 4.2 descreve todas as etapas necessárias para o pré-processamento dos vídeos, incluindo a extração dos *frames* que servem de entrada para uma CNN. A Seção 4.3 descreve o processo de extração das características espaciais, utilizando as camadas convolucionais de uma CNN. Os detalhes sobre a classificação, incluindo o particionamento dos dados, o treinamento da LSTM e a otimização de hiperparâmetros são apresentados na Seção 4.4. Por fim, a Seção 4.5 apresenta o experimento para investigar o impacto da quantidade de *frames* extraídos dos vídeos no resultado do método proposto. A Figura 20 resume as etapas do processo de extração de conhecimento dos vídeos.



Figura 20: Etapas do processo de extração de conhecimento dos vídeos.

Além da descrição do método utilizado, os dados de entrada, os resultados e o classificador proposto são fornecidos aos leitores no repositório de código aberto: https: //github.com/b-mandelbrot/usp-cnn-lstm (acessado em 18 de dezembro de 2021).

4.1 Construção do Conjunto de Dados

Para construir um conjunto de dados foi necessário recorrer a diferentes fontes online, estruturando essas informações. A maior parte dos conjuntos de dados compartilhados com a comunidade científica são de TC e RX. Esse resultado é corroborado pelo levantamento bibliográfico realizado no Capítulo 3, onde os estudos que utilizaram a técnica de USP com foco na COVID-19 foram, em sua grande maioria, realizados com base em apenas dois conjuntos de dados.

A Seção 4.1.1 descreve como foi realizada a pesquisa do conjunto de dados. Na Seção 4.1.2 são apresentados os vídeos utilizados nesta dissertação e na Seção 4.1.3, os dados são caracterizados.

4.1.1 Pesquisa do Conjunto de Dados

A pesquisa do conjunto de dados teve como objetivo encontrar um ou mais repositórios de acesso aberto que pudessem fornecer vídeos de USP para serem utilizados no treinamento e teste de classificadores para auxiliar no diagnóstico da COVID-19.

Por ser uma doença recente, vídeos de USP são mais difíceis de serem encontrados, pois, grande parte desses dados encontra-se distribuída entre diferentes fontes. No entanto, através do levantamento bibliográfico foi possível relacionar dois conjuntos de dados abertos.

O conjunto de dados apresentado em Born, Brändle et al. (2020) (COVID-19 *Lung Ultrasound Dataset*) foi o resultado da coleta de imagens e vídeos de USP em diferentes fontes online, onde os autores compilaram e disponibilizaram em um repositório público todos os dados das outras fontes abertas. Já o conjunto de dados ICLUS-DB: *Italian COVID-19 Lung US Database* foi o resultado da coleta em diversos centros clínicos italianos.

Apesar do ICLUS-DB ser considerado um conjunto de dados aberto e utilizado em algumas publicações (DASTIDER; SADIK; FATTAH, 2021; ROY et al., 2020), o acesso se dá através de uma solicitação de cadastro. Até o momento da escrita deste trabalho, não nos foi concedida autorização de acesso. Dessa forma, a única opção viável foi usar o conjunto de dados COVID-19 *Lung Ultrasound Dataset*. O processo de aquisição dos vídeos deste conjunto de dados é detalhado na Seção 4.1.2 e a caracterização dos dados é descrita na Seção 4.1.3.

4.1.2 Vídeos de Ultrassom Utilizados nesta Dissertação

Devido aos motivos apresentados na Seção 4.1.1, utilizou-se o conjunto de dados COVID-19 *Lung Ultrasound Dataset* para o treinamento e teste dos classificadores em nível de vídeo investigados neste trabalho. Esse conjunto de dados foi construído com base em vídeos de USP, disponibilizados publicamente por diferentes fontes: hospitais, clínicas, fabricantes de equipamentos médicos, publicações científicas, repositórios médicos e plataformas comunitárias. A seleção e validação dos dados foi realizada por dois médicos especialistas na área de US, conforme descrito na Seção 3.2.2.2.

Os vídeos foram adquiridos a partir de transdutores convexos e lineares. Devido à natureza da coleta e dos dados disponibilizados pelas diferentes fontes, não foi possível obter todos os detalhes sobre os fabricantes e modelos dos dispositivos de ultrassom para as aquisições disponibilizadas. No entanto, 11% dos vídeos coletados foram adquiridos por transdutores da fabricante Butterfly modelo iQ+[™] e 25% por meio da fabricante GE Healthcare modelo Venue[™]. Cerca de 50% dos vídeos foram coletados utilizando o protocolo BLUE para aquisição, conforme descrito na Seção 2.1.5.2.

Somente os vídeos adquiridos por transdutores convexos foram utilizados nesta dissertação. Esse tipo de transdutor apresenta menor resolução espacial, mas possui um comprimento de onda acústica mais longa, fornecendo melhor penetração e visualização dos tecidos mais profundos do pulmão (conforme apresentado na Seção 2.1.2.1). Essa característica faz com que esse tipo de transdutor seja mais adequado para avaliar consolidações, derrames pleurais e linhas B (LICHTENSTEIN, 2014; GARCÍA-ARAQUE; ARISTIZÁBAL-LINARES; RUÍZ-ÁVILA, 2015; OLIVEIRA, R. R. de et al., 2020), portanto, são mais versáteis do que os transdutores lineares; sendo encontrados com mais frequência nas instalações médicas e à beira do leito; o que de certa forma justifica a maior quantidade de vídeos adquiridos por meio desse tipo de transdutor no conjunto de dados, conforme explicado em Born, Brändle et al. (2020).

Um *script* na linguagem *Python* foi fornecido com o conjunto de dados, facilitando a transferência de maneira automatizada das diferentes fontes online. Esses vídeos foram armazenados em disco para processamento futuro.

4.1.3 Caracterização do Conjunto de Dados

Ao todo foram obtidos 210 vídeos. No entanto, 22 vídeos adquiridos por transdutores lineares e 3 vídeos de pneumonia viral foram descartados, mantendo a compatibilidade com os modelos preditivos (POCOVID-Net) disponibilizados pelos autores do conjunto de dados em Born, Wiedemann et al. (2021) e que servem de referência para esta dissertação. Por fim, restaram 185 vídeos referentes a 131 pacientes. Um resumo da quantidade de vídeos utilizados por diagnóstico é apresentado na Tabela 2.

Diagnóstico	Quantidade de vídeos	Porcentagem do Total
COVID-19	69	37%
Pneumonia Bacteriana	50	27%
Saudável	66	36%
Total	185	100%

Tabela 2: Quantidade de vídeos de ultrassom incluídos neste estudo.

Em relação aos dados demográficos, apenas 102 vídeos (55,13%) referentes a 67 pacientes (51,14%) continham informações sobre o sexo do paciente. A Figura 21A mostra a ocorrência do sexo dos pacientes nos vídeos e a Figura 21B mostra a ocorrência do sexo por diagnóstico.



Figura 21: Distribuição dos vídeos em relação ao sexo e diagnóstico do paciente.

A distribuição por sexo demonstra uma maior quantidade de vídeos pertencentes

aos pacientes do sexo masculino (Figura 21A). Contudo, para os vídeos com diagnóstico da COVID-19, essa diferença não foi tão significativa, onde 7,84% dos vídeos são de pacientes do sexo feminino e 8,82% do sexo masculino. Para os outros diagnósticos essa diferença foi significativa, onde a quantidade de vídeos de pacientes do sexo masculino foi aproximadamente duas vezes maior do que a do sexo feminino (Figura 21B).

A Figura 22A representa a distribuição da idade dos pacientes nos vídeos. A Figura 22B representa a distribuição da idade por diagnóstico. Cada ponto colorido na imagem representa um paciente submetido ao exame de USP e os diferentes diagnósticos aos quais o exame está relacionado.



Figura 22: Distribuição dos vídeos considerando a idade e diagnóstico do paciente.

As estatísticas relacionadas aos sintomas apresentados pelos pacientes e os achados relacionados aos vídeos de USP são apresentadas na Figura 23. Entretanto, apenas 34% dos vídeos continham informações sobre os sintomas.

A Figura 23A mostra a ocorrência dos sintomas por diagnóstico. A febre foi o sintoma predominantemente relatado para pneumonia bacteriana (81,82%). Para a COVID-19, 52,38% dos casos foram de problemas respiratórios. Em pacientes saudáveis, uma mínima porção apresentava sintomas como fadiga (3,33%), cefaleia (3,33%) e febre (6,67%), mas sem relação com os diagnósticos mencionados anteriormente.

As informações sobre os achados estavam disponíveis para todos os vídeos utilizados neste trabalho. Conforme apresentado na Figura 23B, a maior parte dos achados de pneumonia bacteriana foram de consolidação (71,43%) seguida por derrame pleural (22,45%). Dentre os achados relatados para a COVID-19, encontram-se as linhas B (69,23%), se-



Figura 23: Distribuição dos vídeos em relação a sintomas e achados.

guidas pela linha pleural com irregularidades (41,45%). As linhas A (19,70%) seguidas por um menor número por linhas B (6,06%) foram relatadas nos vídeos referentes aos pacientes saudáveis.

4.2 Pré-processamento dos Dados

A Seção 4.2.1 descreve como os vídeos foram pré-processados, como os artefatos indesejados dos vídeos foram removidos e como os *frames* foram extraídos para processamento. Na Seção 4.2.2 é apresentado o pré-processamento realizado nos *frames*.

4.2.1 Pré-processamento dos Vídeos

Após a coleta dos vídeos diretamente das fontes online, foi aplicado um *script* escrito em *Python* (usando as funções da biblioteca *OpenCV*) fornecido com o conjunto de dados para o pré-processamento dos vídeos. O objetivo foi remover uma série de informações desnecessárias incluindo texto, marcas d'água e marcas registradas do sistema de imagem de US. Os vídeos foram recortados em uma janela quadrada (BORN; BRÄNDLE et al., 2020). Após a remoção das informações, os vídeos foram armazenados em disco utilizando o formato MPEG-4 (do inglês *Moving Picture Experts Group 4*). A Figura 24A apresenta um *frame* de um vídeo contendo informações indesejáveis, as setas vermelhas indicam suas localizações. A Figura 24B representa o resultado do pré-processamento.



Figura 24: Exemplo de informações removidas.

O passo seguinte identificou a quantidade de *frames* disponíveis em cada um dos vídeos coletados. Para essa tarefa foi utilizada a biblioteca OpenCV, disponível para a linguagem *Python*. As estatísticas básicas referentes à análise dos vídeos podem ser verificadas na Tabela 3.

Para que o aproveitamento dos vídeos fosse máximo, optou-se por utilizar o número mínimo de *frames* disponíveis no conjunto de dados (Tabela 3) como um parâmetro a ser considerado na etapa de extração dos *frames*. Todos os vídeos tinham pelo menos 21 *frames*, esse então foi o máximo usado, garantindo que nenhum vídeo fosse descartado.

Apesar de todos os vídeos possuírem um mínimo de 21 *frames*, para que fosse possível verificar o impacto da utilização desses *frames* no desempenho dos classificadores, a

Estatísticas	Frames
Mínimo	21
Mediana	111
Média	148
Máximo	458
Desvio Padrão	100

Tabela 3: Estatísticas sobre a quantidade de frames do conjunto de dados.

extração de *frames* foi separada em quatro configurações: 1) 5 *frames*; 2) 10 *frames*; 3) 15 *frames*; e 4) 20 *frames*, respeitando a quantidade máxima de *frames* estabelecida para uso (21 *frames*).

Foi adotado como limite mínimo o valor de 5 frames (configuração 1) e como limite máximo o valor de 20 frames (configuração 4). As configurações foram padronizadas em intervalos de 5 frames (5, 10, 15 e 20 frames). A extração dos frames se deu em intervalos constantes, com base em cada uma das configurações. Por exemplo, na configuração 1, um vídeo contendo uma sequência de 21 frames (o valor mínimo registrado dentre os 185 vídeos analisados) teve 5 dos 21 frames extraídos — 1 frame extraído a cada 4 frames do vídeo — e o restante dos frames foram descartados (16 frames), conforme apresentado na Figura 25.



Figura 25: Os 5frames extraídos usando a configuração 1.

Para cada vídeo foram extraídas as quantidades de *frames* especificadas em cada uma das 4 configurações (1, 2, 3 e 4). Após a leitura e a escolha dos respectivos *frames*, estes foram extraídos, nomeados e salvos em formato PNG (do inglês *Portable Network Graphics*). A Tabela 4 apresenta a quantidade de *frames* extraídos em cada configuração.

Configuração	Frames	Frames Extraídos
1	5	925
2	10	1850
3	15	2775
4	20	3700

Tabela 4: Quantidade de *frames* extraídos por configuração.

4.2.2 Pré-processamento dos Frames

Cada um dos *frames* armazenados na etapa de pré-processamento dos vídeos (Seção 4.2.1) foi redimensionado, utilizando o algoritmo de interpolação conhecido como *nearest neighbor*. As resoluções encontram-se discriminadas na Tabela 5.

\mathbf{CNN}	Dimensões de Entrada
$\mathrm{DenseNet} 121/169/201$	$224 \times 224 \times 3$
EfficientNetB0	$224 \times 224 \times 3$
InceptionResNetV2	$299\times299\times3$
MobileNetV2	$224 \times 224 \times 3$
NASNetLarge	$331 \times 331 \times 3$
NASNetMobile	$224 \times 224 \times 3$
POCOVID-Net 1,2,3,4,5	$224 \times 224 \times 3$
ResNet152V2	$224 \times 224 \times 3$
VGG16/19	$224 \times 224 \times 3$
Xception	$299\times299\times3$

Tabela 5: Dimensões da camada de entrada para cada uma das CNNs utilizadas.

Esse redimensionamento foi necessário para manter a compatibilidade com as CNNs pré-treinadas no conjunto de dados ImageNet e USP (POCOVID-Net). Após o redimensionamento, foi realizada uma normalização, onde o valor de cada *pixel* foi multiplicado pelo fator de 1/255. No entanto, para a EfficientNetB0 não foi necessário realizar a normalização, devido à existência de uma camada de normalização embutida na própria arquitetura da rede.

Para essa etapa, foi utilizada a classe *ImageDataGenerator* fornecida com o *framework Keras* integrado ao *Tensorflow*. Dessa forma, esse pré-processamento pode ser feito em tempo real para cada *frame* antes da extração de características por uma CNN, etapa que será apresentada na Seção 4.3.

4.3 **Processamento dos Dados**

O processamento dos dados foi caracterizado pela extração das características espaciais realizada por uma CNN. O objetivo foi utilizar as camadas convolucionais desse tipo de rede como um extrator de características (Seção 2.5.1.3). Foram testadas diferentes CNNs, como a POCOVID-Net (BORN; BRÄNDLE et al., 2020), pré-treinada numa versão reduzida desse conjunto de dados e diferentes CNNs pré-treinadas no conjunto de dados ImageNet, conforme a Tabela 5.

Os frames pré-processados utilizando a classe ImageDataGenerator foram submetidos às CNNs selecionadas (Tabela 5) para a extração das características espaciais. Como as redes foram pré-treinadas, não foi necessário retreiná-las para a extração de características. Para não impactar no custo computacional e para que os classificadores pudessem ser comparados (ImageNet versus USP), optou-se por não realizar o ajuste fino (Seção 2.5). Dessa forma, utilizou-se dos padrões aprendidos nesses conjuntos de dados. O procedimento de extração das características é apresentado na próxima seção.

4.3.1 Extração das Características Espaciais

Para a extração das características, as camadas totalmente conectadas das CNNs foram removidas, mantendo-se as camadas convolucionais da rede. Os mapas de características da saída referente à última camada convolucional foram transformados em um vetor de características por meio das camadas de GAP (Seção 2.3.6) ou *flatten* (Seção 2.3.7). Esta última somente utilizada nas VGGs e POCOVID-Nets.

A Figura 26 apresenta um exemplo da saída dos mapas de características referentes as primeiras camadas convolucionais da arquitetura Xception. Cada um dos grupos apresentados na Figura 26A (COVID-19), 26B (Pneumonia Bacteriana) e 26C (Saudável) representam os mapas de características referentes ao primeiro bloco convolucional da Xception, após a camada de convolução (block_1_conv), *batch normalization* (block1_conv1_bn) e ativação (block1_conv1_act). No entanto, antes de serem transformados em vetores de características, esses mapas precisam passar pelas camadas de GPA ou *flatten* (Seção 2.3.7).

A Tabela 6 resume a dimensão dos vetores de características extraídos de cada uma das CNNs utilizadas. Uma vez que esses vetores de características espaciais foram extraídos dos *frames*, eles foram organizados em uma estrutura matricial no formato NPY. O formato NPY é usado para persistir uma matriz em disco, utilizando a biblioteca *Numpy*



Figura 26: As primeiras camadas convolucionais da arquitetura Xception.

para Python. Essa estrutura foi armazenada para inspeção e reutilização, acelerando o processo de treinamento e otimização. Para cada vídeo foram extraídas n sequências de vetores de características espaciais, onde n corresponde a configuração utilizada em cada extração. Por exemplo, para a DensetNet121 utilizando a configuração 1 (5 frames) foi extraída e armazenada uma sequência de 5 vetores contendo 1024 características cada.

4.4 Classificação

Neste trabalho foi utilizada a técnica de validação cruzada com o método conhecido como K-fold. Devido a grande quantidade de classificadores a serem treinados, adotou-se um valor de k = 5, considerando que um valor maior para k implicaria um maior custo computacional, conforme apresentado na Seção 2.7. O particionamento do conjunto de dados pelo método K-fold é apresentado na Seção 4.4.1.

O treinamento da rede recorrente LSTM teve como objetivo aprender a dependência temporal entre os vetores de características espaciais da sequência, atuando como o

CNN	Dimensão do Vetor de Características
DenseNet121	1.024
DenseNet169	1.664
DenseNet201	1.920
EfficientNetB0	1.280
InceptionResNetV2	1.536
MobileNetV2	1.280
NASNetLarge	4.032
NASNetMobile	1.056
POCOVID-Net 1,2,3,4,5	512
ResNet152V2	2.048
VGG16/19	25.088
Xception	2.048

Tabela 6: Dimensão do vetor de características das CNNs.

classificador. O treinamento da LSTM é apresentado na Seção 4.4.2.

4.4.1 Particionamento do Conjunto de Dados

O conjunto de dados foi particionado em 5 partições, cada uma contendo cerca de 20% dos dados. O conjunto de treinamento foi composto pela união de quatro partições ($\approx 80\%$ dos dados) e o conjunto de teste pela partição restante ($\approx 20\%$ dos dados).

Esse procedimento foi repetido cinco vezes, onde a cada iteração o conjunto de testes assumiu uma partição diferente. O particionamento dos dados foi realizado em nível do paciente, não existindo sobreposição de dados no conjunto de treinamento e teste.

Para manter a compatibilidade com o trabalho de referência, foi usado o *script* de código aberto fornecido com o conjunto de dados. Após a execução do *script* de particionamento, obteve-se a distribuição dos dados apresentada na Tabela 7.

Em cada iteração foram treinados e testados cinco classificadores com base na mesma arquitetura (CNN-LSTM) e hiperparâmetros. O desempenho final foi expresso pela média do desempenho de cada um desses classificadores em cada uma das cinco partições avaliadas.

Iteração	Conjunto	COVID-19	Pneumonia Bacteriana	Saudável	Total
1	Treinamento	56	40	53	149
1	Teste	13	10	13	36
2	Treinamento	56	40	53	149
2	Teste	13	10	13	36
3	Treinamento	56	40	52	148
0	Teste	13	10	14	37
4	Treinamento	55	40	53	148
Ĩ	Teste	14	10	13	37
5	Treinamento	53	40	53	146
9	Teste	16	10	13	39

Tabela 7: Distribuição dos vídeos nas partições de treinamento e teste.

4.4.2 Aprendizado das Características Temporais

A LSTM utilizou como entrada uma sequência de vetores de características espaciais extraídos dos *frames* na etapa de extração de características (Seção 4.3.1) e como saída forneceu um dos três diagnósticos (COVID-19, pneumonia bacteriana e saudável). Um diagrama demonstrando a extração e classificação utilizando uma combinação desses dois tipos de redes é apresentado na Figura 27. A Figura 27A representa a sequência de imagens extraídas dos vídeos. A Figura 27B representa as camadas convolucionais de uma VGG16. A Figura 27C representa a camada LSTM responsável por aprender as características temporais. Por fim, a Figura 27D representa as camadas de classificação.

O treinamento da LSTM depende diretamente da configuração adotada na etapa de pré-processamento dos vídeos (Seção 4.2.1) e da arquitetura da CNN utilizada na etapa de extração de características (Seção 4.3.1). Dessa forma, a camada de entrada da LSTM é representada por uma sequência de n vetores de características espaciais.

Ao todo foram implementados 68 classificadores (CNN-LSTM), onde estes foram caracterizados pela combinação entre as configurações de extração de *frames* (1, 2, 3 e 4) e os vetores de características extraídos por cada uma das 17 CNNs utilizadas neste trabalho. A Tabela 8 apresenta um resumo dessas combinações.

A LSTM foi implementada seguindo a mesma quantidade de camadas para todas as 68 combinações de classificadores, portanto, não influenciou no número final de classificadores



Figura 27: Extração e classificação combinando uma VGG e LSTM.

experimentados. Essa arquitetura foi composta de uma camada LSTM utilizando a função de ativação *tanh*, três camadas completamente conectadas, sendo as duas primeiras com função de ativação ReLU e a camada de saída configurada com 3 neurônios (um para cada diagnóstico) e função de ativação *softmax*, esta responsável pelas probabilidades de cada sequência de vetores de características pertencerem a um dos diagnósticos.

CNN	LSTM Input 1	LSTM Input 2	LSTM Input 3	LSTM Input 4
DenseNet121	5×1024	10×1024	15×1024	20×1024
DenseNet169	5×1664	10×1664	15×1664	20×1664
DenseNet201	5×1920	10×1920	15×1920	20×1920
EfficientNetB0	5×1280	10×1280	15×1280	20×1280
InceptionResNetV2	5×1536	10×1536	15×1536	20×1536
MobileNetV2	5×1280	10×1280	15×1280	20×1280
NASNetLarge	5×4032	10×4032	15×4032	20×4032
NASNetMobile	5×1056	10×1056	15×1056	20×1056
POCOVID-Net 1,2,3,4,5	5×512	10×512	15×512	20×512
ResNet152V2	5×2048	10×2048	15×2048	20×2048
VGG16/19	5×25088	10×25088	15×25088	20×25088
Xception	5×2048	10×2048	15×2048	20×2048

Tabela 8: Camada de entrada da LSTM.

Diferentes conjuntos de hiperparâmetros foram variados e testados utilizando um *framework* de otimização. Dentre os hiperparâmetros variados encontram-se: a quantidade de unidades da camada LSTM, a quantidade de neurônios das camadas totalmente conectadas, a taxa de *dropout*, a taxa de aprendizado e o tamanho do lote de treinamento. O procedimento de treinamento e escolha desses hiperparâmetros são descritos na Seção 4.4.3.

4.4.3 Treinamento e Otimização dos Hiperparâmetros dos Classificadores

O treinamento e otimização dos hiperparâmetros dos classificadores foram realizados utilizando o framework Optuna (AKIBA et al., 2019), disponível como uma biblioteca para a linguagem Python. Optuna é um software de código aberto que permite a construção de experimentos para otimização de maneira fácil e organizada, além de fornecer diversos algoritmos para amostragem (sampling) e poda (pruning). É possível executar os experimentos de forma distribuída, e os resultados do processo de otimização são salvos em um banco de dados chamado SQLite, onde os dados podem ser visualizados em um dashboard interativo, através de uma interface web.

Para o amostrador (*sampler*) responsável pela escolha dos hiperparâmetros, foi utilizado o algoritmo TPE (do inglês *Tree-Structured Parzen Estimator*) (BERGSTRA; YA-MINS; COX, 2013) e para o podador (*prunner*) foi adotado o *Hyperband* (LI et al., 2017). A combinação desses algoritimos oferece bons resultados em termos de desempenho e convergência (FALKNER; KLEIN; HUTTER, 2018).

Um total de 68 classificadores foram otimizados, cada classificador correspondendo a uma combinação entre uma das sequências (4 sequências) e uma arquitetura de rede responsável por fornecer um vetor de características espaciais (17 arquiteturas). Por fim, cada classificador foi otimizado por 100 tentativas, onde a cada tentativa o otimizador testou um conjunto de hiperparâmetros, conforme os algoritmos apresentados (Seção 2.10).

Apesar da arquitetura da LSTM não ter variado em função da quantidade de camadas e de outros hiperparâmetros (por exemplo, otimizador e funções de ativação), foi necessário estipular os valores que alguns hiperparâmetros deveriam assumir durante o processo de otimização. Nesse sentido, os hiperparâmetros escolhidos foram: 1) o número de unidades da camada LSTM; 2) a taxa de *dropout*; 3) o número de neurônios nas camadas totalmente conectadas (CTC 1 e CTC 2); 4) a taxa de aprendizado (TA); e 5) o tamanho do lote (TL) a ser utilizado no treinamento. Indiretamente foram testadas cada uma das 4 sequências de entrada, correspondendo as 4 configurações de extração de *frames*, visto que cada um dos classificadores recebeu como entrada uma combinação dessas sequências, conforme Tabela 8. A Tabela 9 apresenta os valores dos hiperparâmetros otimizados para cada um dos 68 classificadores. O treinamento dos classificadores foi realizado por 100 épocas. Foi adotado o otimizador Adam (do inglês Adaptive Moment Estimation), configurado com um beta_1 de 0,9, beta_2 de 0,999 e epsilon de 1×10^{-7} . Além do otimizador Adam, foi utilizada a função de perda de entropia cruzada, porém tanto o otimizador quanto a função de perda não foram variados. O processo de otimização foi realizado considerando a acurácia dos classificadores nas cinco partições, conforme a Seção 2.7. A acurácia média foi utilizada como o valor retornado pela função objetivo a ser maximizada, e os melhores classificadores foram salvos no formato HDF5 (do inglês Hierarchical Data Format 5) para que os desempenhos pudessem ser avaliados.

 $\begin{tabular}{|c|c|c|c|c|c|} \hline Hiperparâmetros & Valores \\ \hline LSTM (unidades) & 32, 64, 128 e 256. \\ CTC 1 (neurônios) & 32, 64 e 128 \\ CTC 2 (neurônios) & 32, 64 e 128 \\ \hline Dropout & 0,1, 0,2, 0,3, 0,4 e 0,5 \\ \hline Taxa de Aprendizado (TA) & 1 \times 10^{-7} a 1 \times 10^{-1} \\ \hline Tamanho do Lote (TL) & 4, 8, 12, 16, 20, 24, 28 e 32 \\ \hline \end{tabular}$

Tabela 9: Valores utilizados para a otimização dos hiperparâmetros.

As bibliotecas *Keras* e *Tensorflow* para a linguagem *Python* foram utilizadas para a implementação das CNNs e LSTMs. O processo de otimização e treinamento foi realizado em uma máquina NVIDIA DGX-1 composta por oito GPUs Tesla P-100 contendo 16 GB de memória cada. No entanto, para este experimento, o processamento foi limitado via código ao consumo de apenas 4096 MB de memória de uma única GPU.

4.5 Impacto das Configurações de Extração dos Frames

Para investigar a existência de uma correlação entre as variáveis, quantidade de frames extraídos dos vídeos (X) e o resultado dos classificadores (Y), foi realizado um teste de correlação. Dentre os testes mais utilizados (Seção 2.11.6), o teste de tau de Kendall foi o escolhido para o experimento, visto que conforme explicado na Seção 2.11.6.1, é um teste não paramétrico, e assim como o teste de Spearman não depende que as variáveis sejam normalmente distribuídas. No entanto, o teste tau de Kendall possui uma vantagem, pois lida com a situação onde existam muitos postos empatados, sendo este o caso. O teste de hipóteses foi construído da seguinte forma:

• $H_0: \tau_B(X,Y) = 0$ (não existe correlação)

- $H_1: \tau_B(X,Y) \neq 0$
- Foi adotado um nível de significância de 5% ($\alpha=0,05)$

Para este experimento foram utilizadas as funções estatísticas fornecidas pela biblioteca *Scipy* (scipy.stats) para a linguagem *Python*.

5 Resultados

Neste capítulo são apresentados os resultados dos experimentos. Na Seção 5.1, são apresentados os resultados do processo de otimização, onde são listados os melhores conjuntos de hiperparâmetros para cada um dos classificadores. Na Seção 5.2, são apresentados os resultados da avaliação dos classificadores com base na técnica de validação cruzada. Na Seção 5.3, são apresentados os resultados do teste de correlação *tau* de Kendall para as configurações de extração de *frames*.

5.1 Treinamento e Otimização dos Hiperparâmetros dos Classificadores

Neste trabalho, foram treinados e otimizados 68 classificadores (CNN-LSTM) para auxiliar no diagnóstico da COVID-19. O método proposto é composto por uma CNN e uma LSTM. As camadas convolucionais da CNN foram utilizadas para extração de características espaciais (Seção 4.3.1) e a LSTM foi responsável pela classificação dos vídeos de USP (Seção 4.4.2).

Ao todo foram selecionadas 17 CNNs pré-treinadas para extração de características, sendo cinco variações da POCOVID-Net (BORN; BRÄNDLE et al., 2020), pré-treinadas em imagens de USP e 12 CNNs pré-treinadas no conjunto de dados ImageNet. Essas CNNs foram combinadas com quatro LSTMs, uma para cada configuração de extração (1, 2, 3 e 4). Cada uma dessas combinações (CNN-LSTM) passou por um processo de otimização de hiperparâmetros, que levou cerca de 75 horas para ser concluído. Os resultados contendo todos os hiperparâmetros selecionados no processo de otimização, para cada uma das combinações são apresentados nas Tabelas 19, 20, 21 e 22 do Apêndice B.

O número de parâmetros treináveis e o tamanho de cada um dos 68 classificadores são apresentados nas Tabelas 23, 24, 25 e 26 do Apêndice C. O número de parâmetros, bem como o tamanho dos classificadores variaram em função da configuração de extração de frames e dos hiperparâmetros selecionados na etapa de otimização.

5.2 Avaliação dos Classificadores

Os resultados da avaliação dos classificadores encontram-se no Apêndice D organizados nas Tabelas 27, 28, 29 e 30. Os resultados da avaliação dos classificadores foram ordenados de forma decrescente pela acurácia, apresentada na primeira coluna das tabelas. Embora essas tabelas listem esses valores com duas casas decimais, para a classificação foram consideradas mais casas decimais em caso de empate, e quando não foi possível o desempate por esse critério, utilizou-se a sensibilidade seguida pela especificidade para a classe COVID-19. Um resumo contendo os 10 melhores classificadores é apresentado na Tabela 10.

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Xception-LSTM	COVID-19	$94,\!44\%$	$100,\!00\%$	$95,\!65\%$	96,77%
(Configuração 4)	Pneumonia	$94{,}55\%$	96,00%	$97{,}69\%$	$95{,}24\%$
Acurácia: 95,00%	Saudável	$97{,}14\%$	$89{,}23\%$	$99{,}13\%$	$92,\!00\%$
DenseNet121-LSTM	COVID-19	$96,\!67\%$	$95,\!38\%$	98,26%	$96,\!00\%$
(Configuração 4)	Pneumonia	$89{,}61\%$	$88,\!00\%$	$97{,}00\%$	$88{,}46\%$
Acurácia: 92,82%	Saudável	$91{,}76\%$	$93{,}85\%$	$93{,}91\%$	$92{,}53\%$
NASNetMobile-LSTM	COVID-19	$91,\!25\%$	$93{,}85\%$	$93,\!91\%$	$92,\!41\%$
(Configuração 3)	Pneumonia	$92{,}50\%$	$90,\!00\%$	$97{,}69\%$	$91,\!11\%$
Acurácia: 92,22%	Saudável	$93,\!33\%$	$92{,}31\%$	$96{,}52\%$	$92{,}80\%$
POCOVID-Net-4-LSTM	COVID-19	90,91%	$89,\!23\%$	$95,\!65\%$	90,00%
(Configuração 2)	Pneumonia	$90{,}91\%$	$92,\!00\%$	$96{,}15\%$	$91{,}43\%$
Acurácia: 92,22%	Saudável	$94{,}29\%$	$95{,}38\%$	$96{,}52\%$	$94{,}81\%$
MobileNetV2-LSTM	COVID-19	$90,\!48\%$	96,92%	$91,\!30\%$	$92,\!94\%$
(Configuração 1)	Pneumonia	$92,\!00\%$	$86,\!00\%$	$98,\!46\%$	88,00%
Acurácia: $91,\!67\%$	Saudável	$94{,}00\%$	$90{,}77\%$	$97{,}39\%$	$92{,}17\%$
DenseNet169-LSTM	COVID-19	91,11%	95,38%	93,04%	92,90%
(Configuração 2)	Pneumonia	$90,\!00\%$	86,00%	$97{,}69\%$	$87,\!50\%$
			Continu	ua na próxii	na página

Tabela 10: Os 10 melhores classificadores.

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 91,67%	Saudável	$93,\!33\%$	$92{,}31\%$	$96{,}52\%$	92,80%
Xception-LSTM	COVID-19	$95,\!00\%$	$89,\!23\%$	98,26%	$91,\!43\%$
(Configuração 1)	Pneumonia	$92,\!00\%$	$92{,}00\%$	$96{,}92\%$	$92{,}00\%$
Acurácia: $91,\!67\%$	Saudável	90,00%	$93{,}85\%$	$92{,}17\%$	$91{,}61\%$
InceptionResNetV2-20	COVID-19	95,56%	$89,\!23\%$	98,26%	$91,\!93\%$
(Configuração 4)	Pneumonia	$89,\!33\%$	$94,\!00\%$	$93{,}85\%$	$91{,}20\%$
Acurácia: 91,13%	Saudável	90,33%	$90{,}77\%$	$94{,}78\%$	$90{,}51\%$
NASNetLarge-LSTM	COVID-19	90,00%	$92,\!31\%$	$93,\!04\%$	$91,\!03\%$
(Configuração 4)	Pneumonia	$90,\!00\%$	$84,\!00\%$	98,46%	$85{,}71\%$
Acurácia: 91,11%	Saudável	$92{,}50\%$	$95,\!38\%$	$94{,}78\%$	$93,\!79\%$
NASNetMobile-LSTM	COVID-19	$90,\!48\%$	$96,\!92\%$	$91,\!30\%$	$92,\!94\%$
(Configuração 2)	Pneumonia	$86,\!67\%$	$84,\!00\%$	$96{,}92\%$	$85{,}00\%$
Acurácia: $90,56\%$	Saudável	$93,\!33\%$	$89,\!23\%$	$97{,}39\%$	$90{,}91\%$

Tabela 10: continuação da página anterior

5.3 Impacto das Configurações de Extração dos Frames

Para verificar o impacto de cada uma das configurações de extração de frames (1, 2, 3 e 4) nos resultados obtidos pelos classificadores, foi realizado um teste não paramétrico de correlação tau de Kendall (Seção 2.11.6.1). Os resultados dos testes (*p*-values) são apresentados na Tabela 11. A rejeição da hipótese nula em um nível de significância de 5% ($\alpha = 0.05$) indica que a correlação existe, $\tau_B \neq 0$.

Tabela 11: Resultados do teste de correlação tau de Kendall.

Classe	Prec.	Sens.	Espec.	F1-Score	Acurácia
COVID-19	$0,\!5263$	0,7664	$0,\!2417$	0,8003	
Pneumonia	$0,\!93$	$0,\!5746$	0,7399	$0,\!6927$	0,5127
Saudável	0,75	0,7915	$0,\!6633$	$0,\!6762$	

Para resumir visualmente os resultados da avaliação dos classificadores nas diferentes configurações $(1, 2, 3 \in 4)$, foi utilizado um gráfico *box plot* combinado a um gráfico de
dispersão (*scatter plot*). O gráfico apresentando na Figura 28 resume os resultados para a acurácia dos classificadores. As Figuras 29, 30 e 31 apresentam os gráficos referentes aos resultados da precisão, sensibilidade, especificidade e F1-*Score* para as diferentes classes de diagnóstico (COVID-19, pneumonia bacteriana e saudável).



Figura 28: Acurácia.



Figura 29: Precisão, sensibilidade, especificidade e F1-Score para a classe COVID-19.



Figura 30: Precisão, sensibilidade, especificidade e F1-*Score* para a classe pneumonia bacteriana.



Figura 31: Precisão, sensibilidade, especificidade e F1-Score para a classe saudável.

6 Discussões

Este capítulo apresenta as discussões dos resultados quanto aos objetivos propostos. Na Seção 6.1, discute-se sobre o melhor classificador e seu conjunto de hiperparâmetros. Na Seção 6.2, são avaliados os resultados obtidos pelos classificadores. Na Seção 6.3 é analisado o impacto do uso das diferentes configurações de extração de *frames* no aprendizado das características temporais. Na Seção 6.4, os resultados obtidos pelos classificadores são comparados com o trabalho de referência. Por fim, na Seção 6.5, são realizadas outras comparações com o método proposto.

6.1 Treinamento e Otimização dos Hiperparâmetros dos Classificadores

De acordo com os resultados obtidos e apresentados na Tabela 10, o melhor classificador segundo o critério adotado de avaliação (acurácia, seguida pela sensibilidade e especificidade para a COVID-19) foi o classificador Xception-LSTM, composto por uma Xception pré-treinada no conjunto de dados ImageNet e uma LSTM configurada com os seguintes hiperparâmetros: 32 unidades LSTM, duas camadas completamente conectadas contendo 128 e 64 neurônios, uma taxa de *dropout* de 0,4 e uma sequência de 20 vetores de características espaciais na camada de entrada (20×2048).

6.2 Avaliação dos Classificadores

Em relação aos resultados numéricos da avaliação, o classificador Xception-LSTM apresentou uma acurácia de 95%, precisão de 94,44%, sensibilidade de 100%, especificidade de 95,65%, e F1-*Score* de 96,77% para a COVID-19. O classificador POCOVID-Net-4-LSTM, cuja base convolucional foi pré-treinada em imagens USP obteve um resultado próximo ao apresentado pelo classificador Xception-LSTM, com uma acurácia média de 92,22%, precisão de 90,91%, sensibilidade de 89,23%, especificidade de 95,65% e F1-*Score* de 90% para COVID-19.

Uma comparação entre os dois classificadores é apresentada na Tabela 12. É perceptível que o classificador Xception-LSTM apresentou melhores resultados para a COVID-19. No entanto, o classificador POCOVID-Net-4-LSTM superou na sensibilidade (95,38%) e no F1-*Score* (94,81%) para a classe saudável. Além disso, também apresentou um resultado similar para a especificidade em relação à COVID-19 (95,65%).

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Xception-LSTM	COVID-19	94,44%	100,00%	95,65%	96,77%
(Configuração 4)	Pneumonia	94,55%	96,00%	97,69%	95,24%
Acurácia: 95,00%	Saudável	97,14%	89,23%	99,13%	92,00%
POCOVID-Net-4-LSTM	COVID-19	90,91%	89,23%	$\begin{array}{c} \textbf{95,65\%}\\ 96,15\%\\ 96,52\%\end{array}$	90,00%
(Configuração 2)	Pneumonia	90,91%	92,00%		91,43%
Acurácia: 92,22%	Saudável	94,29%	95,38%		94,81%

Tabela 12: Resultados da Xception-LSTM e POCOVIDNet-4-LSTM.

Apesar do classificador Xception-LSTM ter apresentado melhores resultados, o classificador POCOVID-Net-4-LSTM utilizou menos *frames*, o que pode ser útil em casos onde os vídeos disponíveis sejam curtos. Outro ponto relevante está relacionado ao número de parâmetros e tamanho dos classificadores. Conforme pode ser observado na Tabela 13, a Xception-LSTM apresentou cerca de 3 vezes menos parâmetros e tamanho do que o classificador POCOVID-Net-4-LSTM.

Tabela 13: Comparação entre Xception-LSTM e POCOVIDNet-4-LSTM.

Posição	Classificador	Parâmetros	Tamanho (MB)
1	Xception-LSTM	279.043	$3,\!23$
4	POCOVIDNet-4-LSTM	837.251	$9,\!62$

Em relação ao número de parâmetros e tamanho dos classificadores, o menor classificador apresentou 73.123 parâmetros e 0,9 MB de tamanho (POCOVIDNet-1-LSTM), porém os resultados numéricos da avaliação foram menores no geral, deixando-o na posição 37 dentre os 68 classificadores avaliados, com uma acurácia de 87,40%. A Tabela 14 apresenta os resultados da avaliação.

Considerando o número de parâmetros e o tamanho dos 10 melhores classificadores, a Xception-LSTM (configuração 4) foi a melhor em relação ao custo benefício. Apresentou um total de 279.043 parâmetros e um tamanho de 3,23 MB. A NASNetMobile-LSTM

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
POCOVID-Net-1-LSTM	COVID-19	$90,\!10\%$	$86,\!44\%$	$95,\!69\%$	$87,\!18\%$
(Configuração 1)	Pneumonia	$89,\!67\%$	$82,\!00\%$	$95,\!46\%$	$85{,}23\%$
Acurácia: 87,40%	Saudável	$84{,}97\%$	$92{,}31\%$	89,77%	$88{,}35\%$

Tabela 14: Resultados da POCOVIDNet-1-LSTM.

(configuração 3) ficou logo em seguida com 293.315 parâmetros e um tamanho de 3,4 MB. No entanto, os resultados da Xception-LSTM são melhores, conforme a Tabela 15.

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Xception-LSTM	COVID-19	94,44%	100,00%	$95,65\%\ 97,69\%\ 99,13\%$	96,77%
(Configuração 4)	Pneumonia	94,55%	96,00%		95,24%
Acurácia: 95,00%	Saudável	97,14%	89,23%		92,00%
NASNetMobile-LSTM	COVID-19	91,25%	93,85%	93,91%	92,41%
(Configuração 3)	Pneumonia	92,50%	90,00%	97,69%	91,11%
Acurácia: 92,22%	Saudável	93,33%	92,31%	96,52%	92,80%

Tabela 15: Resultados da Xception-LSTM e NASNetMobile-LSTM.

6.3 Impacto das Configurações de Extração dos Frames

O número de *frames* utilizados por cada classificador variou segundo as configurações de extração (1, 2, 3 e 4), conforme descrito na Seção 4.3.1. Todas as configurações de extração forneceram bons resultados, dentre os 10 melhores classificadores apresentados na Tabela 10, quatro utilizaram a configuração 4 (20 *frames*), incluindo os dois melhores classificadores.

Segundo os resultados apresentados na Tabela 11, nenhum dos testes de correlação tau de Kendall obteve um valor de *p*-value < 0,05, ou seja, a hipótese nula não foi rejeitada no nível de significância estabelecido, indicando que não existe correlação entre as variáveis (tau = 0). Adicionalmente, os gráficos apresentados nas Figuras 28, 29, 30, e 31 também mostraram ausência de um relacionamento entre as variáveis. Dessa forma, não se pode afirmar que os resultados da avaliação (acurácia, precisão, sensibilidade, especificidade e F1-*Score*) obtidos pelos classificadores estão correlacionados com a quantidade de *frames* extraídos dos vídeos, ou seja, os resultados obtidos pelos classificadores não melhoram nem pioram a medida que utiliza-se mais ou menos *frames*.

6.4 Comparação com o Trabalho de Referência

Em relação ao trabalho usado como referência (BORN; WIEDEMANN et al., 2021), o classificador Xception-LSTM (configuração 4) apresentou melhores resultados que os classificadores propostos como o POCOVID-Net (VGG) e o *Models Genesis* (CNN 3D), conforme apresentado na Seção 3.2.2.2. A versão espaço-temporal da POCOVID-Net (POCOVIDNet-4-LSTM) treinada na configuração 2, obteve uma acurácia de 92% versus 90% da sua versão espacial POCOVID-Net. Além desse desempenho, a versão espaçotemporal POCOVIDNet-4-LSTM obteve um resultado melhor com menos parâmetros (837 K) do que sua versão espacial POCOVID-Net (14,7 M). O classificador *Models Genesis* obteve resultados inferiores a todos os classificadores apresentados, inclusive em relação ao número de parâmetros (7,6 M), salvo quando comparado com a POCOVID-Net (14,7 M).

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Xception-LSTM	COVID-19	94%	100%	96%	97%
(20 frames)	Pneumonia	95%	96%	98%	95%
Acurácia: 95%	Saudável	97%	89%	99%	92%
POCOVID-Net-4-LSTM	COVID-19	$91\% \\ 91\% \\ 94\%$	89%	96%	90%
(10 frames)	Pneumonia		92%	96%	91%
Acurácia: 92%	Saudável		95%	97%	95%
POCOVID-Net (VGG) (5 frames) Acurácia: 90%	COVID-19 Pneumonia Saudável	$92\% \\ 88\% \\ 91\%$	$90\% \\ 93\% \\ 88\%$	96% 95% 95%	$91\% \\ 90\% \\ 89\%$
Models Genesis	COVID-19	77%	74%	$87\% \\ 91\% \\ 88\%$	75%
(12 frames)	Pneumonia	80%	79%		78%
Acurácia: 78%	Saudável	79%	78%		77%

Tabela 16: Resultados da Xception-LSTM, POCOVIDNet-4-LSTM, POCOVID-Net e *Models Genesis.*

6.5 Outras Comparações

O classificador proposto (Xception-LSTM) apresentou melhores resultados que os reportados em Barros et al. (2021), onde foram publicados os resultados preliminares desta dissertação. A Tabela 17 apresenta uma comparação entre os resultados obtidos pelos dois classificadores, mostrando uma melhoria significativa do classificador proposto nesta dissertação. Essa melhoria foi registrada após uma redução na quantidade de valores utilizados no processo de otimização dos hiperparâmetros (Tabela 9.

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Xception-LSTM (20 frames) Acurácia: 95%	COVID-19 Pneumonia Saudável	94% 95% 97%	100% 96% 89%	96% 98% 99%	97% 95% 92%
$Xception-LSTM^1$ (20 frames)	COVID-19 Pneumonia	94% 92%	97% 94%	96 % 96%	95% 93%
Acurácia: 93%	Saudável	95%	89%	98%	91%
(1): Barros et al. (2021)					

Tabela 17: Comparação com os resultados preliminares da Xception-LSTM.

Os resultados também foram comparados com os obtidos por especialistas humanos (em exames de USP), apresentados no estudo realizado em Islam et al. (2021), onde a sensibilidade foi de 86,4%, e a especificidade foi de 54,6% para o diagnóstico da COVID-19 (5 estudos, 446 participantes, 211 casos — 47%). Nesse sentido, o classificador proposto apresentou uma sensibilidade de 100% e uma especificidade de 96%, mostrando potencial para auxiliar no diagnóstico da doença.

Em relação ao protocolo BLUE (apresentado na Seção 2.1.5.3), o classificador apresentou uma acurácia superior (95%) quando comparado ao resultado obtido pelo uso do protocolo (90,5%). No entanto, o protocolo BLUE é mais abrangente e pode diagnosticar outros tipos doenças pulmonares não consideradas pelo classificador proposto.

Outra comparação que pode ser feita está relacionada com o teste RT-PCR. Considerado o padrão ouro para o diagnóstico da COVID-19 (OLIVEIRA, B. A. et al., 2020), este apresenta uma sensibilidade de aproximadamente 70% (WATSON; WHITING; BRUSH, 2020) versus uma sensibilidade de 100% obtida pelo classificador. Entretanto, cabe ressaltar que o classificador proposto é limitado aos exames de USP, o que não ocorre com o teste RT-PCR.

Apesar dos resultados serem positivos, eles não podem ser fidedignamente conclusivos, visto que nesta dissertação são consideradas apenas imagens selecionadas de USP (COVID-19, pneumonia bacteriana e saudáveis). De fato, isso não acontece em cenários reais, onde outras patologias são consideradas, por exemplo: uma pneumonia causada por outro vírus (Influenza), que pode apresentar achados semelhantes e vir a confundir o algoritmo. No entanto, os resultados demonstram que o método proposto tem potencial para auxiliar no diagnóstico de COVID-19 e de outras doenças pulmonares.

7 Conclusões

Este trabalho apresentou um método computacional para a classificação de vídeos de USP para auxiliar no diagnóstico da COVID-19. A proposta de classificação de vídeos de USP envolveu dois tipos de dados: dados espaciais referentes às características extraídas dos *frames* dos vídeos e dados temporais representados pela sequência de vetores de características espaciais. A extração das características espaciais foi realizada por uma CNN, e a dependência temporal entre os *frames* foi aprendida por uma LSTM.

Os resultados indicam que a combinação desses dois tipos de redes neurais (CNN-LSTM) pode ser eficaz no aprendizado de características espaço-temporais, superando o desempenho de métodos com abordagens puramente espaciais, como as que consideram a classificação em nível de *frames* (BORN; WIEDEMANN et al., 2021; AWASTHI et al., 2021). Esses resultados são corroborados por outros estudos, como o apresentado em Donahue et al. (2017), por exemplo. Além disso, o método proposto também superou outras abordagens em nível de vídeo, como as apresentadas em Born, Wiedemann et al. (2021).

A transferência de aprendizagem com CNNs pré-treinadas no conjunto de dados ImageNet forneceu resultados comparáveis com as CNNs pré-treinadas em imagens de USP (em menor número de imagens), sugerindo que o pré-treinamento pode ser utilizado em casos onde existam poucos dados (SHARIF RAZAVIAN et al., 2014). Mostramos também que o uso de técnicas de transferência de aprendizagem e HPO podem ajudar no desenvolvimento de aplicativos para auxiliar no diagnóstico de doenças, como visto em outros estudos (HORRY et al., 2020; LACERDA et al., 2021).

Verificou-se que a quantidade de *frames* extraídos dos vídeos não se relacionou com os resultados obtidos pelos classificadores, ou seja, o fato dessas redes utilizarem mais ou menos *frames* não se correlacionou com os resultados da avaliação, de tal maneira que os classificadores que utilizaram 5 *frames* (representados pelos vetores de características) obtiveram resultados equiparáveis aos obtidos pelos que utilizaram 20 *frames*. Dessa forma, o uso de poucos *frames* pode ser uma alternativa em cenários onde se queira otimizar o tamanho do classificador, a quantidade de parâmetros e, principalmente, quando os vídeos forem muito curtos.

Os resultados obtidos pelo classificador proposto (Xception-LSTM) demonstraram que este pode contribuir na interpretação de USP por especialistas humanos no diagnóstico da COVID-19, melhorando a sensibilidade e especificidade (ISLAM et al., 2021). No entanto, esses resultados devem ser interpretados com cuidado, pois poucos dados estão disponíveis sobre o desempenho de especialistas humanos em imagens de USP para o diagnóstico da COVID-19. Além disso, outros aspectos devem ser considerados, como os apresentados na Seção 7.1, que apresenta as limitações deste trabalho.

Este trabalho forneceu evidências de que a combinação de técnicas de aprendizado profundo e USP podem auxiliar no diagnóstico da COVID-19 e de outras doenças pulmonares, como a pneumonia bacteriana. No entanto, ainda há muito espaço para novos experimentos. Nesse sentido, a Seção 7.2 descreve os trabalhos futuros.

7.1 Limitações do Trabalho

Conforme apresentado na Seção 4.1 o conjunto de dados utilizado neste trabalho apresenta limitações que devem ser consideradas na interpretação dos resultados. Ressalta-se aqui que os dados foram coletados em diferentes fontes públicas, não existindo um padrão em relação às características dos vídeos. Nesse sentido, os vídeos podem não ser fidedignos, visto que não estão em um formato de arquivamento apropriado, como o DICOM.

Não foi possível verificar informações relevantes sobre todos os dispositivos de ultrassom utilizados, como o fabricante, o modelo e a frequência de operação dos transdutores convexos. Além disso, o protocolo utilizado para a realização dos exames não foi especificado para todos os vídeos do conjunto de dados (no entanto, cerca de 50% dos vídeos foram adquiridos utilizando o protocolo BLUE). Também não foi mencionada a experiência do médico ultrassonografista que executou o exame e o método utilizado para confirmar o diagnóstico. Apesar disso, todos os vídeos foram validados por dois médicos especialistas na área de US.

As informações demográficas dos pacientes, como idade e sexo estavam incompletas, conforme analisado na Seção 4.1.3. A idade dos pacientes pode conter um viés, principalmente para os pacientes diagnosticados com a COVID-19. As distribuições dos sintomas, achados radiológicos e das classes de diagnóstico podem não representar a realidade, pois muitos vídeos foram coletados de publicações e repositórios onde o intuito foi o de divulgar determinados achados, portanto, é provável que apresentem alguma incorreção.

Além disso, este trabalho possui uma limitação em relação às classes de diagnóstico, pois ao não considerar outros tipos de doenças pulmonares, pode induzir à classificação incorreta de uma determinada doença, visto que as pneumonias causadas por diferentes tipos de vírus (por exemplo, o vírus Influenza) podem compartilhar os mesmos achados radiológicos (AMORIM et al., 2013; FIOCRUZ, 2020; CAPONE et al., 2020).

Por fim, ressalta-se que os resultados apresentados neste trabalho não foram confirmados em um conjunto de dados independente, dada a dificuldade de acesso a outros conjuntos de dados que pudessem ser utilizados como uma alternativa nesta dissertação.

7.2 Trabalhos Futuros

Apesar dos resultados terem sido positivos, diferentes melhorias podem ser realizadas neste trabalho. Como ponto de partida, deseja-se aumentar o número de vídeos disponíveis, possibilitando testá-los em um conjunto de dados independente, aumentando a generalização dos resultados.

Outro ponto de melhoria está relacionado à expansão do conjunto de dados, pois os experimentos realizados neste trabalho utilizaram vídeos com apenas dois tipos de diagnósticos de doenças pulmonares (COVID-19 e pneumonia bacteriana). Idealmente, para utilização como uma ferramenta de auxílio à detecção e diagnóstico por computador, deve-se considerar uma maior diversidade de vídeos, contendo outros tipos de doenças pulmonares, por exemplo, as apresentadas pelo protocolo BLUE, conforme mencionado na Seção 2.1.5.3. Nesse sentido, a busca por novos dados pode ampliar o número de diagnósticos considerados na classificação.

Em relação ao treinamento, pretende-se verificar se a técnica conhecida como *data* augmentation pode agregar valor aos resultados obtidos neste trabalho, ampliando o número de imagens extraídas dos vídeos. Além disso, pretende-se testar outros tipos de pré-processamento, como o N-CLAHE (KOONSANIT et al., 2017), conforme sugerido em Horry et al. (2020), no intuito de ajudar a realçar pequenos detalhes, texturas e contraste local.

Outro aspecto que pode ser melhorado é que a Xception (extrator de características) não foi treinada em imagens de USP, dessa forma, entende-se que esse treinamento especializado pode ajudar a melhorar a etapa de extração de características (Seção 4.3.1). Adicionalmente, pode-se treinar o classificador de ponta-a-ponta, o que não foi considerado neste trabalho.

Pretende-se realizar novos testes com outros tipos de redes recorrentes, como as GRUs (do inglês *Gated Recurrent Units*). Estas podem ser mais eficientes e, dependendo do conjunto de dados, possuem potencial de fornecer resultados comparáveis ou até melhores do que as LSTMs (CHO et al., 2014).

Por fim, alguns trabalhos consideram a gravidade do acometimento pulmonar causado pela COVID-19 (BALOESCU et al., 2020; DASTIDER; SADIK; FATTAH, 2021; ROY et al., 2020) com base em exames de USP. Esse tipo de informação poderia, por exemplo, ajudar na triagem de pacientes, sendo útil no auxílio a decisões cruciais em momentos de pandemia e, principalmente, para o acompanhamento da doença.

REFERÊNCIAS

AKIBA, T.; SANO, S.; YANASE, T.; OHTA, T.; KOYAMA, M. Optuna: A
Next-Generation Hyperparameter Optimization Framework. In: PROCEEDINGS of the
25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.
Anchorage, AK, USA: Association for Computing Machinery, 2019. (KDD '19),
p. 2623–2631. ISBN 9781450362016. DOI: 10.1145/3292500.3330701.

AKRAM, T.; ATTIQUE, M.; GUL, S.; SHAHZAD, A.; ALTAF, M.; NAQVI, S. S. R.; DAMAŠEVIČIUS, R.; MASKELIŪNAS, R. A novel framework for rapid diagnosis of COVID-19 on computed tomography scans. **Pattern analysis and applications**, Springer, v. 24, n. 3, p. 951–964, jan. 2021. DOI: 10.1007/s10044-020-00950-0.

AMATYA, Y.; RUPP, J.; RUSSELL, F. M.; SAUNDERS, J.; BALES, B.;

HOUSE, D. R. Diagnostic use of lung ultrasound compared to chest radiograph for suspected pneumonia in a resource-limited setting. **International Journal of Emergency Medicine**, Springer London, v. 11, n. 8, mar. 2018. ISSN 18651380. DOI: 10.1186/s12245-018-0170-2.

AMORIM, V. B.; RODRIGUES, R. S.; BARRETO, M. M.; ZANETTI, G.;

MARCHIORI, E. Achados na tomografia computadorizada em pacientes com infecção pulmonar pelo vírus influenza A (H1N1). **Radiologia Brasileira**, Publicação do Colégio Brasileiro de Radiologia e Diagnóstico por Imagem, v. 46, n. 5, p. 299–306, out. 2013. ISSN 1678-7099. DOI: 10.1590/S0100-39842013000500006.

ANGUITA, D.; GHELARDONI, L.; GHIO, A.; ONETO, L.; RIDELLA, S. The "K" in K-fold cross validation. In: PROCEEDINGS of The 20th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium: Louvain-La-Neuve, abr. 2012. p. 441–446. ISBN 9782874190490.

ARNTFIELD, R.; VANBERLO, B.; ALAIFAN, T.; PHELPS, N.; WHITE, M.; CHAUDHARY, R.; HO, J.; WU, D. Development of a convolutional neural network to differentiate among the etiology of similar appearing pathological b lines on lung ultrasound: A deep learning study. **BMJ Open**, BMJ Publishing Group, v. 11, n. 3, mar. 2021. ISSN 20446055. DOI: 10.1136/bmjopen-2020-045120. ASLAN, M. F.; UNLERSEN, M. F.; SABANCI, K.; DURDU, A. CNN-based transfer learning-BiLSTM network: A novel approach for COVID-19 infection detection. **Applied Soft Computing**, v. 98, p. 106912, jan. 2021. ISSN 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2020.106912.

AUJAYEB, A. Could lung ultrasound be used instead of auscultation? African journal of emergency medicine : Revue africaine de la medecine d'urgence, African Federation for Emergency Medicine, v. 10, n. 3, p. 105–106, set. 2020. ISSN 2211-4203. DOI: 10.1016/j.afjem.2020.04.007.

AWASTHI, N.; DAYAL, A.; CENKERAMADDI, L. R.; YALAVARTHY, P. K. Mini-COVIDNet: Efficient Light Weight Deep Neural Network for Ultrasound based Point-of-Care Detection of COVID-19. **IEEE Transactions on Ultrasonics**, **Ferroelectrics, and Frequency Control**, v. 68, n. 6, p. 2023–2037, mar. 2021. ISSN 1525-8955. DOI: 10.1109/TUFFC.2021.3068190.

BAKHRU, R. N.; SCHWEICKERT, W. D. Intensive Care Ultrasound: I. Physics, Equipment, and Image Quality. Annals of the American Thoracic Society, American Thoracic Society, v. 10, p. 540–548, 5 out. 2013. ISSN 23256621. DOI: 10.1513/ANNALSATS.201306-1910T.

BALOESCU, C.; TOPOREK, G.; KIM, S.; MCNAMARA, K.; LIU, R.; SHAW, M. M.; MCNAMARA, R. L.; RAJU, B. I.; MOORE, C. L. Automated Lung Ultrasound B-Line Assessment Using a Deep Learning Algorithm. **IEEE Transactions on Ultrasonics**, **Ferroelectrics, and Frequency Control**, Institute of Electrical e Electronics Engineers Inc., v. 67, n. 11, p. 2312–2320, nov. 2020. ISSN 15258955. DOI: 10.1109/TUFFC.2020.3002249.

BARROS, B.; LACERDA, P.; ALBUQUERQUE, C.; CONCI, A. Pulmonary COVID-19: Learning Spatiotemporal Features Combining CNN and LSTM Networks for Lung Ultrasound Video Classification. **Sensors**, MDPI AG, v. 21, n. 16, p. 5486, ago. 2021. ISSN 14248220. DOI: 10.3390/s21165486.

BERA, S.; SHRIVASTAVA, V. K. Effect of pooling strategy on convolutional neural network for classification of hyperspectral remote sensing images. IET Image
Processing, Institution of Engineering e Technology, v. 14, n. 3, p. 480–486, fev. 2020. ISSN 17519659. DOI: 10.1049/iet-ipr.2019.0561.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. Journal of Machine Learning Research, v. 13, n. 2, p. 281–305, fev. 2012.

BERGSTRA, J.; YAMINS, D.; COX, D. Making a Science of Model Search:Hyperparameter Optimization in Hundreds of Dimensions for Vision Architectures. In:PROCEEDINGS of the 30th International Conference on Machine Learning. Atlanta,Georgia, USA: PMLR, jun. 2013. v. 28. (Proceedings of Machine Learning Research),p. 115–123.

BERRAR, D. Cross-Validation. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics, Academic Press, v. 1-3, p.542–545, jan. 2019. DOI: 10.1016/B978-0-12-809633-8.20349-X.

BHATTACHARYA, S.; REDDY MADDIKUNTA, P. K.; PHAM, Q. V.;
GADEKALLU, T. R.; KRISHNAN S, S. R.; CHOWDHARY, C. L.; ALAZAB, M.;
JALIL PIRAN, M. Deep learning and medical image processing for coronavirus (COVID-19) pandemic: A survey. Sustainable Cities and Society, Elsevier Ltd,
v. 65, p. 102589, fev. 2021. ISSN 22106707. DOI: 10.1016/j.scs.2020.102589.
BORN, J.; BRÄNDLE, G.; COSSIO, M.; DISDIER, M.; GOULET, J.; ROULIN, J.;
WIEDEMANN, N. POCOVID-Net: Automatic Detection of COVID-19 From a New Lung Ultrasound Imaging Dataset (POCUS). [S. l.: s. n.], 2020. arXiv: 2004.12084 [eess.IV].

BORN, J.; WIEDEMANN, N.; COSSIO, M.; BUHRE, C.; BRÄNDLE, G.; LEIDERMANN, K.; AUJAYEB, A.; MOOR, M.; RIECK, B.; BORGWARDT, K. Accelerating detection of lung pathologies with explainable ultrasound image analysis. **Applied Sciences (Switzerland)**, MDPI AG, v. 11, n. 2, p. 1–23, jan. 2021. ISSN 20763417. DOI: 10.3390/app11020672.

BRAHIER, T.; MEUWLY, J.-Y.; PANTET, O.; BROCHU VEZ, M.-J.; GERHARD DONNET, H.; HARTLEY, M.-A.; HUGLI, O.; BOILLAT-BLANCO, N. Lung Ultrasonography for Risk Stratification in Patients with Coronavirus Disease 2019 (COVID-19): A Prospective Observational Cohort Study. **Clinical Infectious Diseases**, set. 2020. ISSN 1058-4838. DOI: 10.1093/cid/ciaa1408.

BUI, A. A.; TAIRA, R. K. Medical Imaging Informatics. Boston, MA: Springer US, 2010. p. 1–446. ISBN 978-1-4419-0384-6. DOI: 10.1007/978-1-4419-0385-3.

BUONSENSO, D.; PATA, D.; CHIARETTI, A. COVID-19 outbreak: less stethoscope, more ultrasound. **The Lancet Respiratory Medicine**, Elsevier, v. 8, n. 5, e27, 2020. DOI: 10.1016/S2213-2600(20)30120-X.

BUSHBERG, J. T.; BOONE, J. M. The essential physics of medical imaging. Philadelphia: Lippincott Williams & Wilkins, 2011. ISBN 1451118104. CAPONE, D.; CAPONE, R.; PEREIRA, A. C. H.; BRUNO, L. P.;

REIS VISCONTI, N. R. G. dos; JANSEN, J. M. Diagnóstico por imagem na pneumonia por COVID-19. **Pulmão RJ**, v. 29, n. 1, p. 22–27, 2020. ISSN 1415-4315. Disponível em: <http://www.sopterj.com.br/wp-

 $\verb|content/themes/_sopterj_redesign_2017/_revista/2017/n_01/12-artigo.pdf>.$

CAPP, E.; NIENOV, O. H. **Bioestatística quantitativa aplicada**. Porto Alegre: UFRGS, 2020. ISBN 9786586232431.

CHEN, T.; WU, D.; CHEN, H.; YAN, W.; YANG, D.; CHEN, G.; MA, K.; XU, D.; YU, H.; WANG, H.; WANG, T.; GUO, W.; CHEN, J.; DING, C.; ZHANG, X.; HUANG, J.; HAN, M.; LI, S.; LUO, X.; ZHAO, J.; NING, Q. Clinical characteristics of 113 deceased patients with coronavirus disease 2019: Retrospective study. **The BMJ**, BMJ Publishing Group, v. 368, mar. 2020. ISSN 17561833. DOI: 10.1136/bmj.m1091.

CHERNIAK, C. The bounded brain: toward quantitative neuroanatomy. Journal of Cognitive Neuroscience, MIT Press, Cambridge, MA, USA, v. 2, n. 1, p. 58–68, 1990. DOI: 10.1162/jocn.1990.2.1.58.

CHO, K.; MERRIËNBOER, B. van; GULCEHRE, C.; BAHDANAU, D.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In: PROCEEDINGS of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, out. 2014. p. 1724–1734. DOI: 10.3115/v1/D14-1179.

CHOLLET, F. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI, USA: IEEE, jul. 2017. 2017-January, p. 1800–1807. ISBN 9781538604571. DOI: 10.1109/CVPR.2017.195. arXiv: 1610.02357.

CLARK, S.; LIU, E.; FRAZIER, P.; WANG, J.; OKTAY, D.; VESDAPUNT, N. **MOE: A global, black box optimization engine for real world metric optimization**. [S. l.: s. n.], 2014. Disponível em https://github.com/Yelp/MOE, acessado em 22 de mar. de 2021.

CONCI, A.; AZEVEDO, E.; LETA, F. R. Computação Gráfica - Vol. 2. Rio de Janeiro: Elsevier, 2008. ISBN 9788535223293.

DASTIDER, A. G.; SADIK, F.; FATTAH, S. A. An integrated autoencoder-based hybrid CNN-LSTM model for COVID-19 severity prediction from lung ultrasound. **Computers in biology and medicine**, Elsevier Ltd, v. 132, February, p. 104296, 2021. ISSN 1879-0534. DOI: 10.1016/j.compbiomed.2021.104296.

DEMI, L. Lung ultrasound: The future ahead and the lessons learned from COVID-19. **The Journal of the Acoustical Society of America**, Acoustical Society of America (ASA), v. 148, n. 4, p. 2146–2150, out. 2020. ISSN 0001-4966. DOI: 10.1121/10.0002183.

DENG, J.; DONG, W.; SOCHER, R.; LI, L.-J.; LI, K.; FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. Miami, FL, USA: Institute of Electrical e Electronics Engineers Inc., jun. 2009. p. 248–255. DOI: 10.1109/CVPR.2009.5206848.

DESAI, S. B.; PAREEK, A.; LUNGREN, M. P. Deep learning and its role in COVID-19 medical imaging. Intelligence-Based Medicine, Elsevier BV, v. 3-4, p. 100013, dez. 2020. ISSN 26665212. DOI: 10.1016/j.ibmed.2020.100013.

DEXHEIMER NETO, F. L.; DALCIN, P. d. T. R.; TEIXEIRA, C.; BELTRAMI, F. G. Ultrassom pulmonar em pacientes críticos: uma nova ferramenta diagnóstica. Jornal Brasileiro de Pneumologia, SciELO Brasil, v. 38, p. 246–256, 2012. DOI: 10.1590/S1806-37132012000200015.

DINH, V. Ultrasound Machine Basics-Knobology, Probes, and Modes - **POCUS 101**. [S. l.: s. n.], set. 2021. Disponível em https:

//www.pocus101.com/ultrasound-machine-basics-knobology-probes-and-modes, acessado em 16 de set. de 2021.

DONAHUE, J.; HENDRICKS, L. A.; ROHRBACH, M.; VENUGOPALAN, S.; GUADARRAMA, S.; SAENKO, K.; DARRELL, T. Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 39, n. 4, p. 677–691, set. 2017. DOI: 10.1109/TPAMI.2016.2599174.

ELHASSOUNY, A.; SMARANDACHE, F. Trends in deep convolutional neural Networks architectures: A review. In: PROCEEDINGS of 2019 International Conference of Computer Science and Renewable Energies, ICCSRE 2019. [S. l.]: Institute of Electrical e Electronics Engineers Inc., jul. 2019. ISBN 9781728108278. DOI: 10.1109/ICCSRE.2019.8807741. ESTEVA, A.; CHOU, K.; YEUNG, S.; NAIK, N.; MADANI, A.; MOTTAGHI, A.; LIU, Y.; TOPOL, E.; DEAN, J.; SOCHER, R. Deep learning-enabled medical computer vision. **npj Digital Medicine**, Nature Research, v. 4, n. 1, p. 1–9, dez. 2021. ISSN 23986352. DOI: 10.1038/s41746-020-00376-2.

FACELI, K.; LORENA, A. C.; GAMA, J.; CARVALHO, A. C. P. d. L. F. d. Inteligência artificial: uma abordagem de aprendizado de máquina. Rio de Janeiro: LTC, ago. 2011. ISBN 9788521618805.

FALKNER, S.; KLEIN, A.; HUTTER, F. BOHB: Robust and Efficient Hyperparameter Optimization at Scale. 35th International Conference on Machine Learning,
ICML 2018, International Machine Learning Society (IMLS), v. 4, p. 2323–2341, jul. 2018. arXiv: 1807.01774.

FERREIRA, J. C.; PATINO, C. M. Understanding diagnostic tests. Part 1. Jornal
Brasileiro de Pneumologia, Sociedade Brasileira de Pneumologia e Tisiologia, v. 43,
p. 330–330, 5 set. 2017. ISSN 18063756. DOI: 10.1590/S1806-37562017000000330.

FEURER, M.; HUTTER, F. Hyperparameter Optimization. In: AUTOMATED Machine Learning: Methods, Systems, Challenges. Cham: Springer International Publishing, 2019. p. 3–33. ISBN 978-3-030-05318-5. DOI: 10.1007/978-3-030-05318-5_1.

FIOCRUZ. COVID-19: orientações da Febrasgo para Atendimento na Gestação, Parto, Puerpério e Abortamento. [S. l.: s. n.], abr. 2020. Disponível em https://portaldeboaspraticas.iff.fiocruz.br/atencao-mulher/covid-19orientacoes-da-febrasgo-para-avaliacao-e-tratamento-ambulatorial-degestantes/, acessado em 20 de out. de 2021.

GARCÍA-ARAQUE, H. F.; ARISTIZÁBAL-LINARES, J. P.; RUÍZ-ÁVILA, H. A. Semiology of lung ultrasonography – Dynamic monitoring available at the patient's bedside. **Colombian Journal of Anesthesiology**, Sociedad Colombiana de Anestesiologia y Reanimacion (SCARE), v. 43, n. 4, p. 290–298, out. 2015. ISSN 22562087. DOI: 10.1016/j.rcae.2015.04.008.

GAYATHRI, J.; ABRAHAM, B.; SUJARANI, M.; NAIR, M. S. A computer-aided diagnosis system for the classification of COVID-19 and non-COVID-19 pneumonia on chest X-ray images by integrating CNN with sparse autoencoder and feed forward neural network. **Computers in Biology and Medicine**, p. 105134, dez. 2021. ISSN 00104825. DOI: 10.1016/j.compbiomed.2021.105134.

GIBBONS, R. C.; MAGEE, M.; GOETT, H.; MURRETT, J.; GENNINGER, J.; MENDEZ, K.; TRIPOD, M.; TYNER, N.; COSTANTINO, T. G. Lung Ultrasound vs. Chest X-ray for the Radiographic Diagnosis of COVID-19 Pneumonia in a High Prevalence Population. **The Journal of Emergency Medicine**, Elsevier BV, p. 615–625, fev. 2021. ISSN 07364679. DOI: 10.1016/j.jemermed.2021.01.041.

GONZALEZ, R. C.; WOODS, R. E. **Processamento de imagens digitais**. Rio de Janeiro: Editora Blucher, 2000. ISBN 9788521202646.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, MA: MIT Press, 2016. ISBN 9780262035613.

GOZES, O.; FRID-ADAR, M.; GREENSPAN, H.; BROWNING, P. D.; ZHANG, H.; JI, W.; BERNHEIM, A.; SIEGEL, E. Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis. [S. l.: s. n.], 2020. arXiv: 2003.05037 [eess.IV].

GU, J.; WANG, Z.; KUEN, J.; MA, L.; SHAHROUDY, A.; SHUAI, B.; LIU, T.; WANG, X.; WANG, G.; CAI, J.; CHEN, T. Recent advances in convolutional neural networks. **Pattern Recognition**, Elsevier Ltd, v. 77, p. 354–377, mai. 2018. ISSN 00313203. DOI: 10.1016/j.patcog.2017.10.013.

HASSAN, H.; REN, Z.; ZHAO, H.; HUANG, S.; LI, D.; XIANG, S.; KANG, Y.; CHEN, S.; HUANG, B. Review and classification of AI-enabled COVID-19 CT imaging models based on computer vision tasks. **Computers in Biology and Medicine**, p. 105123, dez. 2021. ISSN 00104825. DOI: 10.1016/j.compbiomed.2021.105123.

HAYKIN, S. **Redes neurais: princiépios e prática**. 2a. edição. Porto Alegre: Bookman Editora, jan. 2007. ISBN 9788577800865.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: PROCEEDINGS of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. Las Vegas, NV, USA: IEEE, dez. 2016. 2016-December,
p. 770–778. ISBN 9781467388504. DOI: 10.1109/CVPR.2016.90. arXiv: 1512.03385.

HEBB, D. O. The organization of behavior: A neuropsychological theory. Hove, East Sussex, United Kingdom: Psychology Press, jun. 1949. ISBN 9780415654531.

HOCHREITER, S. The vanishing gradient problem during learning recurrent neural nets and problem solutions. International Journal of Uncertainty, Fuzziness and Knowlege-Based Systems, World Scientific Publishing Co. Pte Ltd, v. 6, n. 2, p. 107–116, abr. 1998. ISSN 02184885. DOI: 10.1142/S0218488598000094.

HOCHREITER, S.; SCHMIDHUBER, J. Long Short-Term Memory. Neural Computation, MIT Press, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 08997667. DOI: 10.1162/neco.1997.9.8.1735.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the National Academy of Sciences**, National Academy of Sciences, v. 79, n. 8, p. 2554–2558, abr. 1982. ISSN 0027-8424. DOI: 10.1073/pnas.79.8.2554.

HOPFIELD, J. J. Neurons with graded response have collective computational properties like those of two-state neurons. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 81, n. 10, p. 3088–3092, mai. 1984.

HORRY, M. J.; CHAKRABORTY, S.; PAUL, M.; ULHAQ, A.; PRADHAN, B.; SAHA, M.; SHUKLA, N. COVID-19 Detection through Transfer Learning Using Multimodal Imaging Data. **IEEE Access**, Institute of Electrical e Electronics Engineers Inc., v. 8, p. 149808–149824, ago. 2020. ISSN 21693536. DOI: 10.1109/ACCESS.2020.3016780.

HOWARD, A. G.; ZHU, M.; CHEN, B.; KALENICHENKO, D.; WANG, W.; WEYAND, T.; ANDREETTO, M.; ADAM, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. [S. l.: s. n.], 2017. arXiv: 1704.04861 [cs.CV].

HUANG, G.; LIU, Z.; MAATEN, L. van der; WEINBERGER, K. Q. Densely Connected Convolutional Networks. **Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017**, Institute of Electrical e Electronics Engineers Inc., 2017-January, p. 2261–2269, ago. 2016. arXiv: 1608.06993.

HUANG, Q.; ZHANG, F.; LI, X. Machine Learning in Ultrasound Computer-Aided
Diagnostic Systems: A Survey. BioMed research international, Hindawi Limited,
v. 2018, p. 5137904, mar. 2018. ISSN 2314-6141. DOI: 10.1155/2018/5137904.

HUANG, R.; ZHU, L.; XUE, L.; LIU, L.; YAN, X.; WANG, J.; ZHANG, B.; XU, T.; JI, F.; ZHAO, Y.; CHENG, J.; WANG, Y.; SHAO, H.; HONG, S.; CAO, Q.; LI, C.; ZHAO, X.-a.; ZOU, L.; SANG, D.; ZHAO, H.; GUAN, X.; CHEN, X.; SHAN, C.; XIA, J.; CHEN, Y.; YAN, X.; WEI, J.; ZHU, C.; WU, C. Clinical findings of patients with coronavirus disease 2019 in Jiangsu province, China: A retrospective, multi-center study. Edição: Helton da Costa Santiago. **PLOS Neglected Tropical Diseases**, Public Library of Science, v. 14, n. 5, e0008280, mar. 2020. ISSN 1935-2735. DOI: 10.1371/journal.pntd.0008280.

HUTTER, F.; HOOS, H. H.; LEYTON-BROWN, K. Sequential model-based optimization for general algorithm configuration. In: LECTURE Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Berlin, Heidelberg: Springer, 2011. 6683 LNCS, p. 507–523. ISBN 9783642255656. DOI: 10.1007/978-3-642-25566-3_40.

INSTITUTE OF MEDICINE, N. A. o. S.; ACKERMAN, S. **Discovering the Brain**. Washington, DC: The National Academies Press, 1992. ISBN 978-0-309-46799-5. DOI: 10.17226/1785.

ISLAM, N.; EBRAHIMZADEH, S.; SALAMEH, J.-P.; KAZI, S.; FABIANO, N.; TREANOR, L.; ABSI, M.; HALLGRIMSON, Z.; LEEFLANG, M. M.; HOOFT, L.; POL, C. B. van der; PRAGER, R.; HARE, S. S.; DENNIE, C.; SPIJKER, R.; DEEKS, J. J.; DINNES, J.; JENNISKENS, K.; KOREVAAR, D. A.; COHEN, J. F.; BRUEL, A. V. den; TAKWOINGI, Y.; WIJGERT, J. van de; DAMEN, J. A.; WANG, J.; MCINNES, M. D.; GROUP, C. C.-1. D. T. A. Thoracic imaging tests for the diagnosis of COVID-19. Cochrane Database of Systematic Reviews, John Wiley & Sons, Ltd, v. 2021, n. 3, mar. 2021. ISSN 1465-1858. DOI: 10.1002/14651858.CD013639.PUB4.

JADERBERG, M.; SIMONYAN, K.; ZISSERMAN, A. et al. Spatial transformer networks. Advances in neural information processing systems, v. 2, p. 2017–2025, dez. 2015.

JAMIESON, K.; TALWALKAR, A. Non-stochastic Best Arm Identification and Hyperparameter Optimization. Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS 2016, PMLR, p. 240–248, fev. 2015. arXiv: 1502.07943.

JIMÉNEZ, J.; GINEBRA, J. pyGPGO: Bayesian optimization for Python. Journal of Open Source Software, v. 2, n. 19, p. 431, 2017.

JORDAN, M. I. Chapter 25 - Serial Order: A Parallel Distributed Processing Approach. In: NEURAL-NETWORK Models of Cognition. North-Holland: Elsevier, 1997. v. 121. (Advances in Psychology). p. 471–495. ISBN 9780444819314. DOI: https://doi.org/10.1016/S0166-4115(97)80111-2.

JÚNIOR, D. A. D.; CRUZ, L. B. da; DINIZ, J. O. B.; SILVA, G. L. F. da; JUNIOR, G. B.; SILVA, A. C.; PAIVA, A. C. de; NUNES, R. A.; GATTASS, M. Automatic method for classifying COVID-19 patients based on chest X-ray images, using deep features and PSO-optimized XGBoost. Expert Systems with Applications, Elsevier, v. 183, p. 115452, 2021. DOI: 10.1016/j.eswa.2021.115452.

KATELLA, K. [S. l.: s. n.], dez. 2021. Disponível em https://www.yalemedicine.org/news/covid-19-variants-of-concern-omicron, acessado em 20 de dez. de 2021.

KENDALL, M. G. A new measure of rank correlation. **Biometrika**, v. 30, n. 1-2, p. 81–93, jun. 1938. ISSN 00063444. DOI: 10.1093/biomet/30.1-2.81.

KERAS. Complete guide to transfer learning & fine-tuning in Keras. [S. l.: s. n.], mai. 2020. Disponível em

https://keras.io/guides/transfer_learning/#the-typical-transferlearningworkflow, acessado em 20 de ago. de 2021.

KHAN, A.; SOHAIL, A.; ZAHOORA, U.; QURESHI, A. S. A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, Springer Science e Business Media B.V., v. 53, n. 8, p. 5455–5516, dez. 2020. ISSN 15737462. DOI: 10.1007/s10462-020-09825-6. arXiv: 1901.06032.

KHAN, S.; RAHMANI, H.; SHAH, S. A. A.; BENNAMOUN, M. A guide to convolutional neural networks for computer vision. Synthesis Lectures on Computer Vision, Morgan & Claypool Publishers, v. 8, n. 1, p. 1–207, 2018. DOI: 10.2200/S00822ED1V01Y201712C0V015.

KIAMANESH, O.; HARPER, L.; WISKAR, K.; LUKSUN, W.; MCDONALD, M.; ROSS, H.; WOO, A.; GRANTON, J. Lung Ultrasound for Cardiologists in the Time of COVID-19. **Canadian Journal of Cardiology**, Elsevier Inc., v. 36, n. 7, p. 1144–1147, jul. 2020. ISSN 0828282X. DOI: 10.1016/j.cjca.2020.05.008.

KIM, Y. I.; KIM, S. G.; KIM, S. M.; KIM, E. H.; PARK, S. J.; YU, K. M.;
CHANG, J. H.; KIM, E. J.; LEE, S.; CASEL, M. A. B.; UM, J.; SONG, M. S.;
JEONG, H. W.; LAI, V. D.; KIM, Y.; CHIN, B. S.; PARK, J. S.; CHUNG, K. H.;
FOO, S. S.; POO, H.; MO, I. P.; LEE, O. J.; WEBBY, R. J.; JUNG, J. U.; CHOI, Y. K.
Infection and Rapid Transmission of SARS-CoV-2 in Ferrets. Cell Host and Microbe,
Cell Press, v. 27, n. 5, 704–709.e2, mai. 2020. ISSN 19346069. DOI:
10.1016/j.chom.2020.03.023.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: IJCAI'95: Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2. Montreal, Quebec, Canada: Morgan Kaufmann Publishers Inc., ago. 1995. v. 2, p. 1137–1145. KOONSANIT, K.; THONGVIGITMANEE, S.; PONGNAPANG, N.;

THAJCHAYAPONG, P. Image enhancement on digital x-ray images using n-clahe. In: IEEE. 2017 10th Biomedical Engineering International Conference (BMEiCON). Hokkaido, Japan: IEEE, 2017. p. 1–4. ISBN 9781538608821. DOI: 10.1109/BMEiCON.2017.8229130.

KRIZHEVSKY, A.; HINTON, G. Learning multiple layers of features from tiny images. Toronto, Ontario, 2009.

KULHARE, S.; ZHENG, X.; MEHANIAN, C.; GREGORY, C.; ZHU, M.;
GREGORY, K.; XIE, H.; MCANDREW JONES, J.; WILSON, B. Ultrasound-based detection of lung abnormalities using single shot detection convolutional neural networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 11042
LNCS, January, p. 65–73, 2018. ISSN 16113349. DOI: 10.1007/978-3-030-01045-4_8.
LA ROSA, G.; BONADONNA, L.; LUCENTINI, L.; KENMOE, S.; SUFFREDINI, E.
Coronavirus in water environments: Occurrence, persistence and concentration methods
A scoping review. Water Research, v. 179, p. 115899, jul. 2020. ISSN 00431354.
DOI: 10.1016/j.watres.2020.115899.

LACERDA, P.; BARROS, B.; ALBUQUERQUE, C.; CONCI, A. Hyperparameter Optimization for COVID-19 Pneumonia Diagnosis Based on Chest CT. **Sensors**, v. 21, n. 6, p. 2174, mar. 2021. ISSN 1424-8220. DOI: 10.3390/s21062174.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, IEEE, v. 86, n. 11, p. 2278–2324, nov. 1998. DOI: 10.1109/5.726791.

LEE, W.; ROH, Y. Ultrasonic transducers for medical diagnostic imaging. **Biomedical** engineering letters, Springer, v. 7, n. 2, p. 91–97, mar. 2017. DOI: 10.1007/s13534-017-0021-8.

LI, L.; JAMIESON, K.; DESALVO, G.; ROSTAMIZADEH, A.; TALWALKAR, A.
Hyperband: A Novel Bandit-Based Approach to Hyperparameter Optimization. J.
Mach. Learn. Res., JMLR.org, v. 18, n. 1, p. 6765–6816, jan. 2017. ISSN 1532-4435.
LICHTENSTEIN, D.; GOLDSTEIN, I.; MOURGEON, E.; CLUZEL, P.; GRENIER, P.; ROUBY, J. J. Comparative Diagnostic Performances of Auscultation, Chest
Radiography, and Lung Ultrasonography in Acute Respiratory Distress Syndrome.
Anesthesiology, v. 100, n. 1, p. 9–15, jan. 2004. ISSN 00033022. DOI: 10.1097/0000542-200401000-00006.

LICHTENSTEIN, D. A. Lung ultrasound in the critically ill. Annals of intensive care, SpringerOpen, v. 4, n. 1, p. 1–12, jan. 2014. DOI: 10.1186/2110-5820-4-1.

LICHTENSTEIN, D. A.; MEZIERE, G. A. Relevance of lung ultrasound in the diagnosis of acute respiratory failure: the BLUE protocol. **Chest**, Elsevier, v. 134, n. 1, p. 117–125, 2008. DOI: 10.1378/chest.07-2800.

LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K.; DOLLÁR, P. Focal Loss for Dense Object Detection. In: 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy: IEEE, out. 2017. p. 2999–3007. ISBN 9781538610336. DOI: 10.1109/ICCV.2017.324.

LIU, S.; WANG, Y.; YANG, X.; LEI, B.; LIU, L.; LI, S. X.; NI, D.; WANG, T. Deep learning in medical ultrasound analysis: a review. **Engineering**, Elsevier, v. 5, n. 2, p. 261–275, abr. 2019. ISSN 20958099. DOI: 10.1016/j.eng.2018.11.020.

LIU, X.; FAES, L.; KALE, A. U.; WAGNER, S. K.; FU, D. J.; BRUYNSEELS, A.; MAHENDIRAN, T.; MORAES, G.; SHAMDAS, M.; KERN, C.; LEDSAM, J. R.; SCHMID, M. K.; BALASKAS, K.; TOPOL, E. J.; BACHMANN, L. M.; KEANE, P. A.; DENNISTON, A. K. A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis. **The Lancet Digital Health**, Elsevier, v. 1, n. 6, p.e271–e297, out. 2019. ISSN 2589-7500. DOI: 10.1016/S2589-7500(19)30123-2.

MARCOT, B. G.; HANEA, A. M. What is an optimal value of k in k-fold cross-validation in discrete Bayesian network analysis? **Computational Statistics**, Springer, v. 36, n. 3, p. 2009–2031, jun. 2021. DOI: 10.1007/s00180-020-00999-9.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, Springer, v. 5, n. 4, p. 115–133, 1943.

MCDERMOTT, C.; ŁACKI, M.; SAINSBURY, B.; HENRY, J.; FILIPPOV, M.; ROSSA, C. Sonographic Diagnosis of COVID-19: A Review of Image Processing for Lung Ultrasound. **Frontiers in big data**, v. 4, p. 612561, mar. 2021. ISSN 2624-909X. DOI: 10.3389/fdata.2021.612561.

MCHUGH, M. L. Interrater reliability: the kappa statistic. **Biochemia Medica**, Croatian Society for Medical Biochemistry e Laboratory Medicine, v. 22, n. 3, p. 276, 2012. ISSN 13300962. DOI: 10.11613/bm.2012.031.

MINSKY, M.; PAPERT, S. A. Perceptrons: An introduction to computational geometry. Cambridge, MA, USA: MIT Press, 1969. ISBN 9780262130431.

MITCHELL, C.; RAHKO, P. S.; BLAUWET, L. A.; CANADAY, B.; FINSTUEN, J. A.; FOSTER, M. C.; HORTON, K.; OGUNYANKIN, K. O.; PALMA, R. A.; VELAZQUEZ, E. J. Guidelines for performing a comprehensive transthoracic echocardiographic examination in adults: recommendations from the American Society of Echocardiography. Journal of the American Society of Echocardiography, Elsevier, v. 32, n. 1, p. 1–64, 2019. DOI: 10.1016/j.echo.2018.06.004.

MONGODI, S.; ORLANDO, A.; ARISI, E.; TAVAZZI, G.; SANTANGELO, E.; CANEVA, L.; POZZI, M.; PARIANI, E.; BETTINI, G.; MAGGIO, G.; PERLINI, S.; PREDA, L.; IOTTI, G. A.; MOJOLI, F. Lung Ultrasound in Patients with Acute Respiratory Failure Reduces Conventional Imaging and Health Care Provider Exposure to COVID-19. **Ultrasound in Medicine and Biology**, Elsevier USA, v. 46, n. 8, p. 2090–2093, ago. 2020. ISSN 1879291X. DOI:

10.1016/j.ultrasmedbio.2020.04.033.

MORETTIN, P. A.; BUSSAB, W. O. Estatística básica. 9 ed. São Paulo: Saraiva, 2017. ISBN 9788547220228.

MUHAMMAD, G.; HOSSAIN, M. S. COVID-19 and non-COVID-19 classification using multi-layers fusion from lung ultrasound images. **Information Fusion**, Elsevier, v. 72, p. 80–88, 2021. DOI: 10.1016/j.inffus.2021.02.013.

NAJARIAN, K.; SPLINTER, R. Biomedical signal and image processing. 2nd Edition. Boca Raton, Florida: CRC Press, jun. 2012. ISBN 9781439870334.

NARIN, A.; KAYA, C.; PAMUK, Z. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. **Pattern Analysis and Applications**, v. 24, n. 3, p. 1207–1220, 2021. ISSN 1433-755X. DOI: 10.1007/s10044-021-00984-y.

NETO, M. J. F.; QUEIROZ, M. R. G. de. Rational use of chest ultrasound to confront covid-19. **Radiologia Brasileira**, Colegio Brasileiro de Radiologia, v. 53, n. 5, p. ix–x, set. 2020. ISSN 01003984. DOI: 10.1590/0100-3984.2020.53.5e3.

NGUYEN, T. T.; NGUYEN, Q. V. H.; NGUYEN, D. T.; HSU, E. B.; YANG, S.;

EKLUND, P. Artificial Intelligence in the Battle against Coronavirus (COVID-19): A Survey and Future Research Directions. [S. l.: s. n.], abr. 2021. arXiv: 2008.07343 [cs.CY].

OLIVEIRA, B. A.; OLIVEIRA, L. C. d.; SABINO, E. C.; OKAY, T. S. SARS-CoV-2 and the COVID-19 disease: a mini review on diagnostic methods. **Revista do** Instituto de Medicina Tropical de Sao Paulo, SciELO Brasil, v. 62, 2020. DOI: 10.1590/S1678-9946202062044.

OLIVEIRA, R. Cinesiologia - Estudo do Movimento. [S. l.: s. n.], ago. 2020. Disponível em https:

//ead.gnomio.com/pluginfile.php/743/mod_resource/content/1/CINESIOLOGIA-%20ESTUD0%20D0%20M0VIMENT0.pdf, accessado em 16 de dez. de 2021.

OLIVEIRA, R. R. de; RODRIGUES, T. P.; SILVA, P. S. D. da; GOMES, A. C.; CHAMMAS, M. C. Lung ultrasound: An additional tool in COVID-19. Radiologia Brasileira, Colegio Brasileiro de Radiologia, v. 53, n. 4, p. 241–251, jul. 2020. ISSN 01003984. DOI: 10.1590/0100-3984.2020.0051.

PEIXOTO, A. O.; COSTA, R. M.; UZUN, R.; FRAGA, A. d. M. A.; RIBEIRO, J. D.; MARSON, F. A. L. Applicability of lung ultrasound in COVID-19 diagnosis and evaluation of the disease progression: A systematic review. **Pulmonology**, Elsevier BV, mar. 2021. ISSN 25310437. DOI: 10.1016/j.pulmoe.2021.02.004.

PROMED. Promed Post - ProMED-mail. [S. l.: s. n.], mar. 2021. Disponível em https://promedmail.org/promed-post/?id=6864153, acessado em 22 de mar. de 2021.

PROVOST, F.; FAWCETT, T. Data Science for Business: What you need to know about data mining and data-analytic thinking. Sebastopol, CA: O'Reilly Media, Inc., ago. 2013. ISBN 1449361323.

RESENDE, C. P.; NAVECA, F. G.; LINS, R. D.; ZIMMER DEZORDI, F.; FERRAZ, M. V.; MOREIRA, E. G.; COÊLHO, D. F.; COUTO MOTTA, F.; DIAS PAIXÃO, C. A.; APPOLINARIO, L.; SERRANO LOPES, R.; FONSECA MENDONÇA, A. C. d.; BARRETO DA ROCHA, A. S.; NASCIMENTO, V.; SOUZA, V.; SILVA, G.; NASCIMENTO, F.; GONÇALVES LIMA NETO, L.; RIEDIGER, I.; DO CARMO DEBUR, M.; BRANDAO LEITE, A.; MATTOS, T.; FERNANDES DA COSTA, C.; MOTA PEREIRA, F.; KHOURI, R.; LEAL BERNARDES, F. A.; DELATORRE, E.; GRÄ, T.; MENDONÇA SIQUEIRA, M.; BELLO, G.; WALLAU, G. L. The ongoing evolution of variants of concern and interest of SARS-CoV-2 in Brazil revealed by convergent indels in the amino (N)-terminal domain of the Spike protein. **medRxiv**, Cold Spring Harbor Laboratory Press, p. 2021.03.19.21253946, mar. 2021. DOI: 10.1101/2021.03.19.21253946. ROSENBLATT, F. The perceptron: a probabilistic model for information storage and organization in the brain. **Psychological review**, American Psychological Association, v. 65, n. 6, p. 386, 1958.

ROY, S.; MENAPACE, W.; OEI, S.; LUIJTEN, B.; FINI, E.; SALTORI, C.; HUIJBEN, I.; CHENNAKESHAVA, N.; MENTO, F.; SENTELLI, A.; PESCHIERA, E.; TREVISAN, R.; MASCHIETTO, G.; TORRI, E.; INCHINGOLO, R.; SMARGIASSI, A.; SOLDATI, G.; ROTA, P.; PASSERINI, A.; VAN SLOUN, R. J.; RICCI, E.; DEMI, L. Deep Learning for Classification and Localization of COVID-19 Markers in Point-of-Care Lung Ultrasound. **IEEE Transactions on Medical Imaging**, Institute of Electrical e Electronics Engineers Inc., v. 39, n. 8, p. 2676–2687, ago. 2020. ISSN 1558254X. DOI: 10.1109/TMI.2020.2994459.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, Nature Publishing Group, v. 323, n. 6088, p. 533–536, out. 1986. ISSN 00280836. DOI: 10.1038/323533a0.

RUMELHART, D. E.; MCCLELLAND, J. L.; PDP RESEARCH GROUP, C. (Ed.). Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations. Cambridge, MA, USA: MIT Press, 1986. ISBN 026268053X.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, Springer New York LLC, v. 115, n. 3, p. 211–252, set. 2014. DOI: 10.1007/s11263-015-0816-y. arXiv: 1409.0575.

SABINO, E. C.; BUSS, L. F.; CARVALHO, M. P.; PRETE, C. A.; CRISPIM, M. A.;
FRAIJI, N. A.; PEREIRA, R. H.; PARAG, K. V.; DA SILVA PEIXOTO, P.;
KRAEMER, M. U.; OIKAWA, M. K.; SALOMON, T.; CUCUNUBA, Z. M.;
CASTRO, M. C.; DE SOUZA SANTOS, A. A.; NASCIMENTO, V. H.;
PEREIRA, H. S.; FERGUSON, N. M.; PYBUS, O. G.; KUCHARSKI, A.;
BUSCH, M. P.; DYE, C.; FARIA, N. R. Resurgence of COVID-19 in Manaus, Brazil,
despite high seroprevalence. Elsevier B.V., v. 397, n. 10273, p. 452–455, fev. 2021. ISSN 1474547X. DOI: 10.1016/S0140-6736(21)00183-5.

SANDLER, M.; HOWARD, A.; ZHU, M.; ZHMOGINOV, A.; CHEN, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. **Proceedings of the IEEE** Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, p. 4510–4520, jan. 2018. DOI: 10.1109/CVPR.2018.00474. arXiv: 1801.04381.

SARVAMANGALA, D. R.; KULKARNI, R. V. Convolutional neural networks in medical image understanding: a survey. **Evolutionary Intelligence**, Springer Science e Business Media Deutschland GmbH, v. 1, p. 1–22, jan. 2021. ISSN 1864-5909. DOI: 10.1007/s12065-020-00540-3.

SCHERER, D.; MÜLLER, A.; BEHNKE, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In: ARTIFICIAL Neural Networks – ICANN 2010. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. p. 92–101. ISBN 9783642158254. DOI: 10.1007/978-3-642-15825-4_10.

SHARIF RAZAVIAN, A.; AZIZPOUR, H.; SULLIVAN, J.; CARLSSON, S. CNN
Features Off-the-Shelf: An Astounding Baseline for Recognition. In: PROCEEDINGS of the IEEE conference on computer vision and pattern recognition workshops. Columbus, OH, USA: IEEE, jun. 2014. p. 512–519. ISBN 978-1-4799-4308-1. DOI: 10.1109/CVPRW.2014.131.

SHAW, J. A.; LOUW, E. H.; KOEGELENBERG, C. F. Lung Ultrasound in COVID-19: Not Novel, but Necessary. Respiration, Karger Publishers, v. 99, p. 545–547, 2020. DOI: 10.1159/000509763.

SHERSTINSKY, A. Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) Network. **Physica D: Nonlinear Phenomena**, Elsevier B.V., v. 404, ago. 2018. DOI: 10.1016/j.physd.2019.132306. arXiv: 1808.03314.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. In: 3RD International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings. San Diego, CA, USA: International Conference on Learning Representations, ICLR, mai. 2015. arXiv: 1409.1556.

SLOUN, R. J. van; DEMI, L. Localizing B-lines in lung ultrasonography by weakly supervised deep learning, in-vivo results. **IEEE journal of biomedical and health informatics**, IEEE, v. 24, n. 4, p. 957–964, ago. 2019. DOI: 10.1109/JBHI.2019.2936151.

SOLDATI, G.; SMARGIASSI, A.; INCHINGOLO, R.; BUONSENSO, D.; PERRONE, T.; BRIGANTI, D. F.; PERLINI, S.; TORRI, E.; MARIANI, A.; MOSSOLANI, E. E. Proposal for International Standardization of the Use of Lung Ultrasound for Patients With COVID-19. Journal of Ultrasound in Medicine, John Wiley & Sons, Ltd, v. 39, p. 1413–1419, 7 jul. 2020. ISSN 15509613. DOI: 10.1002/JUM.15285.

SRIVASTAVA, N.; HINTON, G.; KRIZHEVSKY, A.; SUTSKEVER, I.; SALAKHUTDINOV, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, v. 15, n. 56, p. 1929–1958, jan. 2014.

SUETENS, P. Ultrasound Imaging. In: FUNDAMENTALS of Medical Imaging. 3. ed. Cambridge: Cambridge University Press, jul. 2017. p. 147–183. DOI: 10.1017/9781316671849.007.

SWAPNAREKHA, H.; BEHERA, H. S.; NAYAK, J.; NAIK, B. Role of intelligent computing in COVID-19 prognosis: A state-of-the-art review. **Chaos, Solitons and Fractals**, Elsevier Ltd, v. 138, p. 109947, set. 2020. ISSN 09600779. DOI: 10.1016/j.chaos.2020.109947.

SYEDA, H. B.; SYED, M.; SEXTON, K. W.; SYED, S.; BEGUM, S.; SYED, F.; PRIOR, F.; YU, F. Role of machine learning techniques to tackle the covid-19 crisis: Systematic review. **JMIR Medical Informatics**, JMIR Publications Inc., v. 9, n. 1, e23811, jan. 2021. ISSN 22919694. DOI: 10.2196/23811.

SZEGEDY, C.; IOFFE, S.; VANHOUCKE, V.; ALEMI, A. A. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: 31ST AAAI Conference on Artificial Intelligence, AAAI 2017. San Francisco, California, USA: AAAI press, fev. 2017. p. 4278–4284. arXiv: 1602.07261.

SZEGEDY, C.; VANHOUCKE, V.; IOFFE, S.; SHLENS, J.; WOJNA, Z. Rethinking the Inception Architecture for Computer Vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, jun. 2016. p. 2818–2826. ISBN 9781467388511. DOI: 10.1109/CVPR.2016.308.

TAN, M.; LE, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. **36th International Conference on Machine Learning, ICML 2019**, International Machine Learning Society (IMLS), 2019-June, p. 10691–10700, mai. 2019. arXiv: 1905.11946.

TAYARANI N., M. H. Applications of artificial intelligence in battling against covid-19: A literature review. **Chaos, Solitons and Fractals**, Elsevier Ltd, v. 142, p. 110338, jan. 2021. ISSN 09600779. DOI: 10.1016/j.chaos.2020.110338. TSAI, C. H.; BURGT, J. van der; VUKOVIC, D.; KAUR, N.; DEMI, L.; CANTY, D.;
WANG, A.; ROYSE, A.; ROYSE, C.; HAJI, K.; DOWLING, J.; CHETTY, G.;
FONTANAROSA, D. Automatic deep learning-based pleural effusion classification in lung ultrasound images for respiratory pathology diagnosis. Physica Medica:
European Journal of Medical Physics, Associazione Italiana di Fisica Medica, v. 83, p. 38–45, mar. 2021. ISSN 1724191X. DOI: 10.1016/j.ejmp.2021.02.023.
TUNG-CHEN, Y.; GRACIA, M. M. de; DIÉEZ-TASCÓN, A.;
ALONSO-GONZÁLEZ, R.; AGUDO-FERNÁNDEZ, S.; PARRA-GORDO, M. L.;
OSSABA-VÉLEZ, S.; RODRIÉGUEZ-FUERTES, P.; LLAMAS-FUENTES, R.
Correlation between chest computed tomography and lung ultrasonography in patients with coronavirus disease 2019 (COVID-19). Ultrasound in medicine & biology, Elsevier, v. 46, n. 11, p. 2918–2926, nov. 2020. DOI:
10.1016/j.ultrasmedbio.2020.07.003.

VAIDYANATHAN, G. Coronavirus variants are spreading in India - what scientists know so far. **Nature**, mai. 2021. ISSN 1476-4687. DOI: 10.1038/d41586-021-01274-7.

VOLZ, E.; MISHRA, S.; CHAND, M.; BARRETT, J. C.; JOHNSON, R.;
HOPKINS, S.; GANDY, A.; RAMBAUT, A.; FERGUSON, N. M. Transmission of SARS-CoV-2 Lineage B.1.1.7 in England: Insights from linking epidemiological and genetic data. medRxiv, Cold Spring Harbor Laboratory Press, p. 2020.12.30.20249034, jan. 2021. DOI: 10.1101/2020.12.30.20249034.

WALDEN, A.; SMALLWOOD, N.; DACHSEL, M.; MILLER, A.; STEPHENS, J.; GRIKSAITIS, M. Thoracic ultrasound: It's not all about the pleura. **BMJ Open Respiratory Research**, BMJ Publishing Group, v. 5, n. 1, p. 354, out. 2018. ISSN 20524439. DOI: 10.1136/bmjresp-2018-000354.

WANG, K.; GAO, X.; ZHAO, Y.; LI, X.; DOU, D.; XU, C.-Z. Pay Attention to Features, Transfer Learn Faster CNNs. In: INTERNATIONAL Conference on Learning Representations. Virtual Conference, Formerly Addis Ababa ETHIOPIA: [s. n.], abr. 2020.

WANG, S. H.; NAYAK, D. R.; GUTTERY, D. S.; ZHANG, X.; ZHANG, Y. D. COVID-19 classification by CCSHNet with deep fusion using transfer learning and discriminant correlation analysis. **Information Fusion**, Elsevier B.V., v. 68, p. 131–148, abr. 2021. ISSN 15662535. DOI: 10.1016/j.inffus.2020.11.005.

WATSON, J.; WHITING, P. F.; BRUSH, J. E. Interpreting a covid-19 test result. **BMJ**, BMJ Publishing Group Ltd, v. 369, n. 8245, mai. 2020. DOI: 10.1136/bmj.m1808.

WHO. WHO Coronavirus (COVID-19) Dashboard. [S. l.: s. n.], nov. 2021. Disponível em https://covid19.who.int, acessado em 16 de dez. de 2021.

WIKIPEDIA. Multipolar neuron. [S. l.: s. n.], mai. 2021. Disponível em https://en.wikipedia.org/wiki/Multipolar_neuron, acessado em 16 de dez. de 2021.

WU, X.; CHEN, C.; ZHONG, M.; WANG, J.; SHI, J. COVID-AL: The Diagnosis of COVID-19 with Deep Active Learning. **Medical Image Analysis**, Elsevier, p. 101913, nov. 2020. ISSN 13618415. DOI: 10.1016/j.media.2020.101913.

YAGER, R. R.; RYBALOV, A. Uninorm aggregation operators. Fuzzy sets and systems, Elsevier, v. 80, n. 1, p. 111–120, mai. 1996. DOI: 10.1016/0165-0114(95)00133-6.

YANG, P. C.; LUH, K. T.; CHANG, D. B.; WU, H. D.; YU, C. J.; KUO, S. H. Value of sonography in determining the nature of pleural effusion: analysis of 320 cases. **American Journal of Roentgenology**, v. 159, n. 1, p. 29–33, jul. 1992. ISSN 0361803X. DOI: 10.2214/ajr.159.1.1609716.

ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. Dive into Deep Learning.
[S. l.: s. n.], jun. 2021. Disponível em https://d2l.ai, acessado em 16 de set. de 2021.
arXiv: 2106.11342 [cs.LG].

ZHANG, J.; CHNG, C.-B.; CHEN, X.; WU, C.; ZHANG, M.; XUE, Y.; JIANG, J.;
CHUI, C.-K. Detection and Classification of Pneumonia from Lung Ultrasound Images.
In: 2020 5th International Conference on Communication, Image and Signal Processing (CCISP). Chengdu, China: IEEE, nov. 2020. p. 294–298. ISBN 978-1-7281-8589-7. DOI: 10.1109/CCISP51026.2020.9273469.

ZHOU, S. K.; GREENSPAN, H.; DAVATZIKOS, C.; DUNCAN, J. S.;
GINNEKEN, B. van; MADABHUSHI, A.; PRINCE, J. L.; RUECKERT, D.;
SUMMERS, R. M. A review of deep learning in medical imaging: Imaging traits,
technology trends, case studies with progress highlights, and future promises.
Proceedings of the IEEE, Institute of Electrical e Electronics Engineers Inc.,
p. 820–838, ago. 2020. DOI: 10.1109/JPR0C.2021.3054390.

ZHOU, Z.; SODHA, V.; RAHMAN SIDDIQUEE, M. M.; FENG, R.; TAJBAKHSH, N.; GOTWAY, M. B.; LIANG, J. Models genesis: generic autodidactic models for 3d medical image analysis. In: MEDICAL Image Computing and Computer Assisted Intervention – MICCAI 2019. Shenzhen, China: Springer Nature Switzerland, out. 2019. 11767 LNCS. (Lecture Notes in Computer Science), p. 384–393. ISBN 9783030322502. DOI: 10.1007/978-3-030-32251-9_42. arXiv: 1908.06912.

ZHU, F.; ZHAO, X.; WANG, T.; WANG, Z.; GUO, F.; XUE, H.; CHANG, P.; LIANG, H.; NI, W.; WANG, Y.; CHEN, L.; JIANG, B. Ultrasonic Characteristics and Severity Assessment of Lung Ultrasound in COVID-19 Pneumonia in Wuhan, China: A Retrospective, Observational Study. **Engineering**, Elsevier BV, v. 7, n. 3, p. 367–375, mar. 2020. ISSN 20958099. DOI: 10.1016/j.eng.2020.09.007.

ZHU, N.; ZHANG, D.; WANG, W.; LI, X.; YANG, B.; SONG, J.; ZHAO, X.;
HUANG, B.; SHI, W.; LU, R.; NIU, P.; ZHAN, F.; MA, X.; WANG, D.; XU, W.;
WU, G.; GAO, G. F.; TAN, W. A Novel Coronavirus from Patients with Pneumonia in China, 2019. New England Journal of Medicine, Massachusetts Medical Society,
v. 382, n. 8, p. 727–733, fev. 2020. ISSN 0028-4793. DOI: 10.1056/nejmoa2001017.

APÊNDICE A - Resumo dos Trabalhos Relacionados

Este apêndice apresenta a Tabela 18 contendo um resumo do Capítulo 3, onde são apresentados os trabalhos relacionados. Os três primeiros trabalhos são referentes à classificação de *frames* e vídeos no contexto de doenças pulmonares, conforme a Seção 3.1. O restante dos trabalhos são referentes à classificação de *frames* e vídeos no contexto da COVID-19 (Seção 3.2 e seguem a mesma ordem apresentada no Capítulo 3.

Trabalho	Tipo Técnica	Conjunto de Dados	Achados ou Doença	Avaliação (%)
Kulhare et al. (2018)	SSD Frame (Inception V2)	In vivo Transdutor Convexo Modo B 2200 vídeos de suídos 100 exames de US	1 - linhas A 2 - linhas B 3 - linhas B coalescentes 4 - consolidação 5 - derrame pleural 6 - linha pleural	 Linhas A Sensibilidade: 87,2 Especificidade: 89 Linhas B Sensibilidade: 28; Especificidade: 93 Linhas B Coalescentes Sensibilidade: 85 Especificidade: 96,5 Consolidação Sensibilidade: 93,6 Especificidade: 93,6 Especificidade: 86,3 Derrame Pleural Sensibilidade: 87,5; Especificidade: 87,5; Especificidade: 92,2 Linha Pleural Sensibilidade: 85,6 Especificidade: 93,1 Acurácia: 85+

Tabela 18: Resumo dos trabalhos relacionados.

Continua na próxima página...

Trabalho	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
		CNN (Inception V3)	Modo M (simulado)	Pneumotórax (deslizamento pulmonar)	Sensibilidade: 93 Especificidade: 93 Acurácia: 85+
Sloun e Demi (2019)	Frame	CNN	In vitro Transdutor linear Modo B 10 vídeos In vivo Transdutor linear Modo B 15 vídeos 10 pacientes Toshiba (The Aplio XV) In vivo Transdutor linear Modo B 12 vídeos 10 pacientes	Linhas B	In vitro Acuracia: 91,7 Sensibilidade: 91,8 Especificidade: 91,8 VPN: 95 VPP: 86,4 In vivo Acuracia: 83,9 Sensibilidade: 78,6 Especificidade: 86,8 VPN: 88,2 VPP: 76,3 Toshiba (The Aplio XV) In vivo Acuracia: 89,2 Sensibilidade: 87,1 Especificidade: 93 VPN: 79,8 VPP: 95,8
Baloescu et al. (2020)	Vídeo	CNN 3D (12 frames)	Transdutor linear, convexo e phased array	Linhas B	Sensibilidade: 93 Especificidade: 96 AUC: 97 Kappa: 88
			Modo B 400 vídeos 400 pacientes	Pontuação de (0–4) representando o acometimento pulmonar pela presença de linhas B.	Kappa: 65
Roy et al. (2020)	Frame	CNN + Reg-STN + SORD	Transdutor linear e convexo Modo B 277 vídeos 35 pacientes	Pontuação de (0–3) representando a gravidade do acometimento pulmonar causado pela COVID-19.	F1-Score: 65,1

Tabela 18: continuação da página anterior

Trabalho	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
	Vídeo		Transdutor linear e convexo Modo B 60 vídeos 35 pacientes		F1-Score: 61
Dastider, Sadik e Fattah (2021)	Frame	Autocodificadores + Bloco de convolução separável de profundidade + DensetNet-201	Transdutor linear e convexo Modo B 60 vídeos 29 pacientes	Pontuação de (0–3) representando a gravidade do acometimento pulmonar causado pela COVID-19.	Transdutor Linear Acurácia: 70 Sensibilidade.: 70 Especificidade: 90,8 F1-Score: 70,2 Transdutor Convexo Acurácia: 61 Sensibilidade: 61 Especificidade: 75,6 F1-Score: 58,6
	Vídeo	Autocodificadores + Bloco de convolução separável de profundidade + DensetNet-201 + LSTM			Transdutor Linear Acurácia: 79,1 Sensibilidade: 79,1 Especificidade: 90,1 F1-Score: 78,6 Transdutor Convexo Acurácia: 67,7 Sensibilidade: 67,7 Especificidade: 76,8 F1-Score: 66,6

Tabela 18: continuação da página anterior

Continua na próxima página...

Trabalho	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
Horry et al. (2020)	Frame	VGG/19	Transdutor linear e convexo Modo B 911 frames	Grupo 1 COVID-19 e Pneumonia Bacteriana versus Saudável Grupo 2 COVID-19 versus Pneumonia Bacteriana	Grupo 1 COVID-19 e Pneumonia Sensibilidade: 97 Precisão: 99 F1-Score: 98 Normal Sensibilidade: 98 Precisão: 94 F1-Score: 96 Grupo 2 COVID-19 Sensibilidade: 100 Precisão: 100 F1-Score: 100 Pneumonia Bacteriana Sensibilidade: 100 Precisão: 100 F1-Score: 100
Born, Wiedemann et al. (2021)	Frame	POCOVID-Net (VGG)	Transdutor linear e convexo Modo B 202 vídeos 59 imagens 216 pacientes	COVID-19 Pneumonia Bacteriana Saudável	Acurácia: 87,8 COVID-19 Sensibilidade: 88 Precisão: 90 F1-Score: 89 Especificidade: 94 Pneumonia Bacteriana Sensibilidade: 90 Precisão: 81 F1-Score: 85 Especificidade: 94 Saudável Sensibilidade: 83 Precisão: 90 F1-Score.: 86 Especificidade: 94

Tabela 18: continuação da página anterior
Trabalho	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
					Acurácia: 90
					COVID-19
					Sensibilidade: 90
					Precisão: 92
					F1-Score: 91
					Especificidade: 96
					Pneumonia Bacteriana
	Vídeo				Sensibilidade: 93
					Precisão: 88
					F1-Score: 90
					Especificidade: 95
					Saudável
					Sensibilidade: 88
					Precisão: 91
					F1-Score.: 89
					Especificidade: 95
					Acurácia: 93
					COVID-19
					Sensibilidade: 97
					Precisão: 94
					F1-Score: 95
					Especificidade: 96
			Transdutor linear		
Barros et al.	17/1	Xception	Modo B	COVID-19	Pneumonia Bacteriana
(2021)	video		185 vídeos	Pneumonia Bacteriana	Sensibilidade: 94
		LSIM	131 pacientes	Saudavel	Fiecisao: 92
					F 1-Score: 95 Especificidade: 96
					Especificidade. 90
					Saudável
					Sensibilidade: 89
					Precisão: 95
					F1-Score.: 91
					Especificidade: 98
				Co	ntinua na próxima página

Tabela 18: continuação da página anterior

Trabalho 7	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
					Acurácia: 83,2
					COVID-19 Sensibilidade.: 92 Precisão: 83 F1-Score: 87 Especificidade: 71
Awasthi et al. F (2021)	Tame	e Mini-COVIDNet (Perda Focal)	e convexo Modo B 64 vídeos	COVID-19 Pneumonia Bacteriana Saudável	Pneumonia Bacteriana Sensibilidade: 82 Precisão: 92 F1-Score: 87 Especificidade: 97
					Saudável Sensibilidade: 51 Precisão: 70 F1-Score.: 59 Especificidade: 96
					Acurácia: 92,5 AUC: 99,93
Muhammad	Frame Fusão de Dados	CNN com	Transdutor linear e	COVID-19	COVID-19 Sensibilidade: 90,2 Precisão: 95,2
e Hossain F		Modo B 121 vídeos	Pneumonia Bacteriana Saudável	Pneumonia Bacteriana Sensibilidade: 95,8 Precisão: 96,9	
					Saudável Sensibilidade: 93,6 Precisão: 83,2

Tabela 18: continuação da página anterior

Continua na próxima página...

Trabalho	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
Jiaqi Zhang et al. (2020)	Frame	EfficientNet-B5	Modo B 10350 frames	 8 tipos de características clínicas (0-7), são elas: 0 - normal. 1 - quantidade de linhas B inferior a 3. 2 - quantidade de linhas B superior à 3. 3 - área de fusão da linha B é inferior à metade. 4 - área de fusão da linha B é superior à metade. 5 - profundidade das peças é inferior a 1cm. 6 - broncograma aéreo e a profundidade de hepatização são inferiores a 3cm. 7 - derrame pleural e profundidade de hepatização é superior a 3cm. Grupo 1 (0, 1-4 e 5-7) Grupo 2 (0, 1-4, 5-6 e 7) Grupo 3 (0-7) 	Grupo 1 (0, 1–4 e 5–7) F1-Score: 93,2 Acurácia: 95,5 Sensibilidade: 93,2 Especificidade: 96,6 Precisão: 93,2 Grupo 2 (0, 1–4, 5–6 e 7) F1-Score: 89,9 Acurácia: 95 Sensibilidade: 89,9 Especificidade: 96,6 Precisão: 89,9 Grupo 3 (0–7) F1-Score: 81,6 Acurácia: 95,4 Sensibilidade: 81,6 Especificidade: 97,4 Precisão: 81,6
Tsai et al. (2021)	Frame	CNN + Reg-STN proposta em Roy et al. (2020)	Transdutor phased array Modo B 623 vídeos	Derrame Pleural	Acuracia: 92,38 F1-Score: 34,98–90,47 Precisão: 42,82–92,76 Sensibilidade: 29,75–88,24
	Vídeo				Acurácia: 91,12 F1-Score: 84,58–95,68 Precisão: 38,85–87,29 Sensibilidade: 41,26–88,14

Tabela 18: continuação da página anterior

Continua na próxima página...

Trabalho	Tipo	Técnica	Conjunto de Dados	Achados ou Doenças	Avaliação
					COVID-19 com SDRA Sensibilidade: 92,4 Especificidade: 88,3 Precisão: 71,3 F1-Score: 80,5 AUC: 96,5
Arntfield et al. (2021)	Frame	Xception	Transdutor phased array Modo B 612 vídeos 243 pacientes	COVID-19 com SDRA Não COVID-19 com SDRA EPH	Não COVID-19 com SDRA Sensibilidade: 76 Especificidade: 81,5 Precisão: 73,1 F1-Score: 74,6 AUC: 89,3
					EPH Sensibilidade: 69,3 Especificidade: 99,9 Precisão: 99,6 F1-Score: 81,7 AUC: 99,1
					COVID-19 com SDRA Sensibilidade: 100 Especificidade: 92,9 Precisão: 85,7 F1-Score: 92,3 AUC: 100
	Vídeo				Não COVID-19 com SDRA Sensibilidade: 85,7 Especificidade: 76,9 Precisão: 66,7 F1-Score: 75 AUC: 93,4
					EPH Sensibilidade: 57,1 Especificidade: 100 Precisão: 100 F1-Score: 72,7 AUC: 100

Tabela 18: continuação da página anterior

APÊNDICE B - Hiperparâmetros dos Modelos

As Tabelas 19, 20, 21 e 22 apresentam os valores dos melhores hiperparâmetros. Cada tabela está associada a uma configuração de extração de *frames* (Seção 4.3.1) e ordenada de forma crescente pela coluna Posição, representando a posição do classificador segundo os critérios de acurácia, seguida pela sensibilidade e especificidade para a COVID-19. A coluna LSTM representa a quantidade de unidades da camada LSTM; a coluna *Dropout* representa a taxa de *dropout*; as colunas CTC 1 e CTC 2 representam a quantidade de neurônios das camadas completamente conectadas 1 e 2; a coluna TA representa a taxa de aprendizado; e a coluna TL representa o tamanho do lote.

Posição	Classificador	LSTM	Dropout	CTC 1	CTC 2	TA	\mathbf{TL}
5	MobileNetV2-LSTM	256	0,1	64	32	$5,\!46 imes10^{-5}$	12
7	Xception-LSTM	256	0,4	32	128	$3,08 \times 10^{-3}$	8
17	POCOVID-Net-3-LSTM	32	0,1	64	64	$6,58 \times 10^{-3}$	20
24	POCOVID-Net-2-LSTM	128	0,2	64	128	$1,\!49 \times 10^{-3}$	8
25	POCOVID-Net-4-LSTM	256	0,2	64	32	$1,59 imes10^{-3}$	24
27	NasNetMobile-LSTM	64	0,3	32	32	$1,\!24 \times 10^{-3}$	24
34	ResNet152V2-LSTM	128	0,2	32	64	$3,\!48 imes 10^{-4}$	24
35	NasNetLarge-LSTM	64	0,1	32	32	$2,03 \times 10^{-4}$	8
37	POCOVID-Net-1-LSTM	32	0,2	32	64	$9,81 \times 10^{-4}$	28
39	EfficientNetB0-LSTM	64	0,5	64	128	$5,06 imes 10^{-4}$	4
41	DenseNet201-LSTM	256	0,3	64	32	$6,11 \times 10^{-3}$	16
46	POCOVID-Net-5-LSTM	256	0,4	32	64	$1,51 \times 10^{-3}$	12
48	InceptionResNetV2-LSTM	256	0,3	128	32	$4,\!21 imes 10^{-5}$	28
55	DenseNet 121-LSTM	256	0,2	32	32	$6,\!44 imes 10^{-4}$	28
57	VGG19-LSTM	256	0,2	64	64	$4,77 imes 10^{-6}$	28
58	DenseNet169-LSTM	32	0,3	64	32	$1,52 \times 10^{-4}$	24
60	VGG16-LSTM	128	$0,\!4$	128	64	$1{,}02\times10^{-3}$	28

Tabela 19: Hiperparâmetros da configuração 1 (5 frames).

Posição	Classificador	LSTM	Dropout	CTC 1	CTC 2	TA	\mathbf{TL}
4	POCOVID-Net-4-LSTM	256	0.2	128	128	1.73×10^{-3}	12
6	DenseNet169-LSTM	128	0,2	32	128	$2,81 \times 10^{-4}$	20
10	NasNetMobile-LSTM	64	$0,\!5$	32	128	$1,23 \times 10^{-3}$	20
13	EfficientNetB0-LSTM	64	0,3	32	64	$6,24 \times 10^{-5}$	4
14	InceptionResNetV2-LSTM	64	0,3	128	32	$3,12 \times 10^{-4}$	16
18	POCOVID-Net-2-LSTM	32	0,1	128	128	$2,92 \times 10^{-3}$	8
20	DenseNet201-LSTM	256	0,2	128	128	$5,\!23 imes 10^{-4}$	8
26	ResNet152V2-LSTM	32	0,1	128	32	$1,76 imes10^{-4}$	28
28	POCOVID-Net-3-LSTM	256	0,5	32	128	$2,\!19 imes10^{-3}$	24
36	MobileNetV2-LSTM	32	0,4	128	64	$1,\!60 imes 10^{-4}$	12
38	DenseNet121-LSTM	128	0,1	32	32	$1,49 \times 10^{-4}$	32
43	NasNetLarge-LSTM	64	0,5	32	128	$3,46 \times 10^{-3}$	28
44	POCOVID-Net-5-LSTM	256	0,4	32	64	$6,74 imes 10^{-4}$	4
56	Xception-LSTM	32	0,4	128	128	$2,93 \times 10^{-3}$	12
61	VGG16-LSTM	128	0,1	128	128	$8,62 \times 10^{-5}$	28
63	VGG19-LSTM	128	0,1	128	64	$7,\!57 imes10^{-5}$	12
68	POCOVID-Net-1-LSTM	256	0,2	128	64	$1,\!37 imes 10^{-3}$	16

Tabela 20: Hiperparâmetros da configuração 2 (10 frames).

Tabela 21: Hiperparâmetros da configuração 3 (15 frames).

Posição	Classificador	LSTM	Dropout	CTC 1	CTC 2	TA	\mathbf{TL}
3	NasNetMobile-LSTM	64	0,2	64	32	$3,44 \times 10^{-4}$	4
11	InceptionResNetV2-LSTM	128	0,3	128	128	$9,06 \times 10^{-5}$	8
19	Xception-LSTM	64	0,1	32	32	$3,\!17 \times 10^{-4}$	28
22	VGG16-LSTM	256	0,5	64	128	$3,94 \times 10^{-4}$	8
23	NasNetLarge-LSTM	32	0,4	32	64	$6,10 \times 10^{-3}$	24
29	ResNet152V2-LSTM	128	0,2	128	128	$2,\!28 imes 10^{-4}$	4
31	DenseNet201-LSTM	32	0,1	64	128	$6,\!29 imes 10^{-4}$	20
45	POCOVID-Net-2-LSTM	64	0,1	64	32	$8,42 \times 10^{-3}$	12
47	MobileNetV2-LSTM	256	0,3	32	32	$1,90 \times 10^{-4}$	16
49	POCOVID-Net-1-LSTM	256	0,2	32	64	$1,09 \times 10^{-3}$	20
50	POCOVID-Net-5-LSTM	128	0,4	128	32	$4,25 \times 10^{-3}$	12
53	DenseNet 121-LSTM	64	0,2	128	128	$1,59 \times 10^{-4}$	4
54	POCOVID-Net-4-LSTM	64	0,1	128	64	$2,88 \times 10^{-3}$	12
59	EfficientNetB0-LSTM	64	0,4	128	32	$1,\!27 imes 10^{-2}$	8
62	DenseNet169-LSTM	128	0,4	64	128	$1,\!38 imes 10^{-4}$	4
64	POCOVID-Net-3-LSTM	128	0,2	32	128	$1{,}64\times10^{-3}$	32
65	VGG19-LSTM	128	0,1	128	128	$2{,}75\times10^{-4}$	8

Tabela 22: Hiperparâmetros da configuração 4 (20 frames).

Posição	Classificador	LSTM	Dropout	CTC 1	CTC 2	TA	\mathbf{TL}
1	Xception-LSTM	32	$0,\!4$	128	64	$1{,}43\times10^{-3}$	16
2	DenseNet 121-LSTM	256	0,2	64	32	$3,25 \times 10^{-3}$	20
8	InceptionResNetV2-LSTM	256	0,5	64	128	$2,02 \times 10^{-4}$	20
9	NasNetLarge-LSTM	256	0,4	128	64	$6,97 imes 10^{-5}$	12
12	ResNet152V2-LSTM	64	0,3	32	128	$2,66 \times 10^{-4}$	32
15	POCOVID-Net-4-LSTM	256	0,2	64	64	$1,97 imes 10^{-4}$	4
16	NasNetMobile-LSTM	64	0,5	64	32	$6,54 imes10^{-5}$	32
21	VGG16-LSTM	256	0,4	128	64	$2,\!19 imes10^{-3}$	8
30	DenseNet 169-LSTM	256	0,1	32	32	$7,\!49 imes 10^{-3}$	32
32	POCOVID-Net-2-LSTM	256	0,2	64	64	$3,69 \times 10^{-4}$	8
33	EfficientNetB0-LSTM	64	0,1	32	64	$7,\!69 imes 10^{-5}$	28
40	POCOVID-Net-1-LSTM	128	0,2	128	64	$4,39 imes 10^{-3}$	32
42	DenseNet201-20	32	0,5	32	32	$3,\!17 \times 10^{-3}$	28
51	MobileNetV2-LSTM	32	0,1	64	32	$4,50 \times 10^{-4}$	8
52	POCOVID-Net-5-LSTM	64	0,2	32	32	$2,79 imes 10^{-3}$	20
66	VGG19-LSTM	64	0,2	32	64	$4,57 imes 10^{-4}$	4
67	POCOVID-Net-3-LSTM	256	0,2	32	128	$7{,}49\times10^{-5}$	28

APÊNDICE C – Número de Parâmetros e Tamanho dos Classificadores

As Tabelas 23, 24, 25 e 26 apresentam o número de parâmetros e o tamanho dos classificadores em megabytes (MB). Cada tabela está associada a uma configuração de extração de *frames* (Seção 4.3.1) e ordenada de forma crescente pelo número de parâmetros. A coluna Parâmetros representa tanto o total de parâmetros do classificador quanto o total de parâmetros treináveis, pois, possuem o mesmo valor; a coluna Posição representa a posição do classificador segundo os critérios de acurácia, seguida pela sensibilidade e especificidade para a COVID-19; e a coluna Tamanho representa o tamanho do arquivo no formato HDF5 em megabytes (MB) do classificador.

Posição	Classificador	Parâmetros	Tamanho (MB)
37	POCOVID-Net-1-LSTM	73.123	0,9
17	POCOVID-Net-3-LSTM	76.227	$0,\!94$
58	DenseNet169-LSTM	221.507	2,58
27	NasNetMobile-LSTM	290.211	3,36
24	POCOVID-Net-2-LSTM	345.155	$3,\!99$
39	EfficientNetB0-LSTM	357.187	$4,\!13$
55	DenseNet121-LSTM	599.075	6,9
46	POCOVID-Net-5-LSTM	797.987	$9,\!17$
25	POCOVID-Net-4-LSTM	806.083	9,27
35	NasNetLarge-LSTM	1.052.067	12,08
34	ResNet152V2-LSTM	1.121.059	$12,\!87$
5	MobileNetV2-LSTM	1.592.515	$18,\!27$
48	InceptionResNetV2-LSTM	1.845.411	21,16
41	DenseNet201-LSTM	2.247.875	25,77
7	Xception-LSTM	2.373.155	27,2
60	VGG16-LSTM	12.936.067	$148,\!08$
57	VGG19-LSTM	25.974.083	$297,\!29$

Tabela 23: Parâmetros e classificadores da configuração 1 (5 frames).

Posição	Classificador	Parâmetros	Tamanho (MB)
18	POCOVID-Net-2-LSTM	90.883	1,08
36	MobileNetV2-LSTM	180.739	$2,\!11$
26	ResNet152V2-LSTM	274.819	$3,\!19$
56	Xception-LSTM	287.491	3,33
10	NasNetMobile-LSTM	293.667	3,4
13	EfficientNetB0-LSTM	348.707	4,03
14	InceptionResNetV2-LSTM	422.403	$4,\!87$
38	DenseNet121-LSTM	595.619	$6,\!86$
44	POCOVID-Net-5-LSTM	797.987	$9,\!17$
28	POCOVID-Net-3-LSTM	800.291	9,2
68	POCOVID-Net-1-LSTM	828.803	9,53
4	POCOVID-Net-4-LSTM	837.251	$9,\!62$
6	DenseNet169-LSTM	926.755	$10,\!65$
43	NasNetLarge-LSTM	1.055.523	$12,\!12$
20	DenseNet201-LSTM	2.279.043	26,12
63	VGG19-LSTM	12.936.067	148,08
61	VGG16-LSTM	12.944.515	148,18

Tabela 24: Parâmetros e classificadores da configuração 2 (10 frames).

Tabela 25: Parâmetros e classificadores da configuração 3 (15 frames).

Posição	Classificador	Parâmetros	Tamanho (MB)
45	POCOVID-Net-2-LSTM	154.051	1,8
54	POCOVID-Net-4-LSTM	164.483	1,92
31	DenseNet201-LSTM	260.803	3,03
3	NasNetMobile-LSTM	293.315	$3,\!4$
53	DenseNet121-LSTM	304.003	$3,\!52$
64	POCOVID-Net-3-LSTM	336.931	3,9
50	POCOVID-Net-5-LSTM	348.931	4,03
59	EfficientNetB0-LSTM	356.867	$4,\!12$
23	NasNetLarge-LSTM	523.683	6,03
19	Xception-LSTM	544.163	$6,\!27$
49	POCOVID-Net-1-LSTM	797.987	$9,\!17$
11	InceptionResNetV2-LSTM	885.891	$10,\!18$
62	DenseNet169-LSTM	934.979	10,74
29	ResNet152V2-LSTM	1.148.035	$13,\!18$
47	MobileNetV2-LSTM	1.583.267	18,16
65	VGG19-LSTM	12.944.515	148,18
22	VGG16-LSTM	25.978.435	$297,\!34$

Tabela 26: Parâmetros e classificadores da configuração 4 (20 frames).

Posição	Classificador	Parâmetros	Tamanho (MB)
52	POCOVID-Net-5-LSTM	150.947	1,77
51	MobileNetV2-LSTM	172.355	2,01
42	DenseNet201-20	252.195	2,93
1	Xception-LSTM	279.043	$3,\!23$
33	EfficientNetB0-LSTM	348.707	4,03
40	POCOVID-Net-1-LSTM	353.155	4,08
12	ResNet152V2-LSTM	547.619	6,31
67	POCOVID-Net-3-LSTM	800.291	9,2
15	POCOVID-Net-4-LSTM	808.259	$9,\!29$
32	POCOVID-Net-2-LSTM	808.259	$9,\!29$
2	DenseNet 121-LSTM	1.330.371	$15,\!27$
21	VGG16-LSTM	1.385.859	15,9
8	InceptionResNetV2-LSTM	1.861.187	$21,\!34$
30	DenseNet169-LSTM	1.976.483	$22,\!66$
9	NasNetLarge-LSTM	4.433.283	50,78
66	VGG19-LSTM	6.443.555	73,78
16	NasNetMobile-LSTM	6.445.507	73,8

APÊNDICE D - Resultado da Avaliação dos Classificadores

As Tabelas 27, 28, 29 e 30 apresentam os resultados numéricos da avaliação dos classificadores. Cada tabela está associada a uma configuração de extração de *frames* (Seção 4.3.1) e ordenada de forma decrescente segundo os critérios de acurácia, seguida pela sensibilidade e especificidade para a COVID-19.

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
MobileNetV2-LSTM	COVID-19	$90,\!48\%$	96,92%	$91,\!30\%$	$92,\!94\%$
(Configuração 1)	Pneumonia	$92,\!00\%$	$86,\!00\%$	$98{,}46\%$	88,00%
Acurácia: 91,67%	Saudável	$94{,}00\%$	$90{,}77\%$	$97{,}39\%$	$92{,}17\%$
Xception-LSTM	COVID-19	$95,\!00\%$	89,23%	98,26%	$91,\!43\%$
(Configuração 1)	Pneumonia	$92,\!00\%$	$92,\!00\%$	$96{,}92\%$	$92,\!00\%$
Acurácia: 91,67%	Saudável	90,00%	$93{,}85\%$	$92{,}17\%$	$91{,}61\%$
POCOVID-Net-3-LSTM	COVID-19	89,52%	$95,\!38\%$	$90,\!43\%$	$91,\!76\%$
(Configuração 1)	Pneumonia	$90,\!00\%$	88,00%	$96,\!92\%$	$88,\!89\%$
Acurácia: $90,00\%$	Saudável	$91{,}43\%$	$86{,}15\%$	$97{,}39\%$	$88,\!00\%$
POCOVID-Net-2-LSTM	COVID-19	89,41%	90,77%	$92,\!17\%$	$89,\!87\%$
(Configuração 1)	Pneumonia	$89,\!09\%$	$92,\!00\%$	$95{,}41\%$	$90,\!48\%$
Acurácia: $89,46\%$	Saudável	90,00%	$86{,}15\%$	$96{,}52\%$	$87{,}62\%$
POCOVID-Net-4-LSTM	COVID-19	90,00%	$95,\!38\%$	$90,\!43\%$	$91,\!77\%$
(Configuração 1)	Pneumonia	$80,\!00\%$	80,00%	$97{,}69\%$	$80,\!00\%$
			Continua	a na próxin	na página

Tabela 27: Resultados da configuração 1 (5 frames).

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 89,44%	Saudável	$91,\!30\%$	90,77%	$95{,}65\%$	90,93%
NasNetMobile-LSTM	COVID-19	89,23%	89,23%	$93,\!91\%$	$89,\!23\%$
(Configuração 1)	Pneumonia	$86,\!67\%$	88,00%	$93,\!85\%$	$87,\!27\%$
Acurácia: $89{,}44\%$	Saudável	$92{,}73\%$	$90{,}77\%$	$96{,}52\%$	$91{,}67\%$
ResNet152V2-LSTM	COVID-19	87,50%	$93,\!85\%$	$86,\!96\%$	89,73%
(Configuração 1)	Pneumonia	$80,\!00\%$	$86,\!00\%$	$97{,}69\%$	$87,\!50\%$
Acurácia: $88,\!33\%$	Saudável	90,00%	$84{,}62\%$	$97{,}39\%$	$86,\!32\%$
NasNetLarge-LSTM	COVID-19	90,00%	90,77%	$93,\!04\%$	$90,\!23\%$
(Configuração 1)	Pneumonia	$86{,}67\%$	$84,\!00\%$	$96{,}92\%$	$85,\!00\%$
Acurácia: 88,33%	Saudável	$87{,}14\%$	$89{,}23\%$	$92{,}17\%$	$88,\!15\%$
POCOVID-Net-1-LSTM	COVID-19	$90,\!10\%$	$86,\!44\%$	$95{,}69\%$	$87,\!18\%$
(Configuração 1)	Pneumonia	$89{,}67\%$	$82,\!00\%$	$95{,}46\%$	$85{,}23\%$
Acurácia: 87,40%	Saudável	$84{,}97\%$	$92{,}31\%$	$89{,}77\%$	$88,\!35\%$
EfficientNetB0-LSTM	COVID-19	$91,\!43\%$	$86,\!15\%$	$97,\!39\%$	88,00%
(Configuração 1)	Pneumonia	$80,\!00\%$	$80,\!00\%$	$96{,}92\%$	$80,\!00\%$
Acurácia: 87,22%	Saudável	$87,\!20\%$	$93{,}85\%$	86,09%	$89,\!47\%$
DenseNet201-LSTM	COVID-19	$89,\!93\%$	$83,\!08\%$	$95,\!65\%$	86,02%
(Configuração 1)	Pneumonia	$86{,}85\%$	$88,\!00\%$	$94{,}64\%$	$86,\!99\%$
Acurácia: 86,74%	Saudável	$86{,}47\%$	$89{,}23\%$	89,57%	$87,\!36\%$
POCOVID-Net-5-LSTM	COVID-19	85,71%	$87{,}91\%$	$90,\!47\%$	$86,\!48\%$
(Configuração 1)	Pneumonia	$89{,}29\%$	$80,\!00\%$	$97{,}72\%$	$82{,}04\%$
Acurácia: 85,62%	Saudável	$85{,}38\%$	$87,\!80\%$	$89,\!60\%$	86, 36%
InceptionResNetV2-LSTM	COVID-19	$86,\!45\%$	$88,\!09\%$	$91,\!34\%$	$87,\!16\%$
(Configuração 1)	Pneumonia	$88,\!00\%$	$80,\!00\%$	$95{,}46\%$	$83,\!42\%$
Acurácia: 85,12%	Saudável	82,09%	86,15%	$90,\!47\%$	$83,\!86\%$
DenseNet121-LSTM	COVID-19	84,13%	89,52%	87,90%	86,26%
(Configuração 1)	Pneumonia	80,95%	70,00%	$95,\!38\%$	72,64%
			Continua	a na próxin	na página

Tabela 27: continuação da página anterior

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 80,75%	Saudável	$78{,}85\%$	80,00%	$87,\!09\%$	$78{,}54\%$
VGG19-LSTM	COVID-19	$79{,}59\%$	$92,\!99\%$	$84,\!46\%$	$85,\!42\%$
(Configuração 1)	Pneumonia	$100,\!00\%$	50,00%	$100,\!00\%$	$64,\!80\%$
Acurácia: 80,42%	Saudável	$77,\!11\%$	90,77%	$84{,}76\%$	$83,\!23\%$
DenseNet169-LSTM	COVID-19	$81,\!18\%$	83,87%	$84,\!38\%$	$81,\!55\%$
(Configuração 1)	Pneumonia	$77{,}14\%$	$72{,}00\%$	$92{,}52\%$	$74{,}21\%$
Acurácia: 79,68%	Saudável	$79{,}96\%$	$81{,}54\%$	$92{,}25\%$	$79{,}91\%$
VGG16-LSTM	COVID-19	$75{,}21\%$	$79,\!37\%$	$81,\!85\%$	$76{,}13\%$
(Configuração 1)	Pneumonia	57,78%	$48,\!00\%$	97,77%	$52,\!26\%$
Acurácia: 73,61\%	Saudável	$74{,}30\%$	$87{,}69\%$	$79{,}48\%$	$79{,}75\%$

Tabela 27: continuação da página anterior

Tabela 28: Resultados da configuração 2 (10 frames).

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
POCOVID-Net-4-LSTM	COVID-19	$90,\!91\%$	89,23%	$95,\!65\%$	$90,\!00\%$
(Configuração 2)	Pneumonia	90,91%	$92,\!00\%$	$96,\!15\%$	$91,\!43\%$
Acurácia: $92,22\%$	Saudável	$94{,}29\%$	$95{,}38\%$	$96{,}52\%$	$94{,}81\%$
DenseNet169-LSTM	COVID-19	91,11%	$95,\!38\%$	93,04%	$92,\!90\%$
(Configuração 2)	Pneumonia	$90,\!00\%$	$86,\!00\%$	$97{,}69\%$	$87,\!50\%$
Acurácia: 91,67%	Saudável	$93,\!33\%$	$92{,}31\%$	$96{,}52\%$	$92{,}80\%$
NasNetMobile-LSTM	COVID-19	90,48%	96,92%	$91,\!30\%$	$92,\!94\%$
(Configuração 2)	Pneumonia	$86,\!67\%$	$84,\!00\%$	96,92%	$85,\!00\%$
Acurácia: 90,56%	Saudável	$93,\!33\%$	$89,\!23\%$	$97{,}39\%$	90,91%
EfficientNetB0-LSTM	COVID-19	86,55%	92,31%	$90,\!47\%$	$89,\!19\%$
(Configuração 2)	Pneumonia	$93,\!33\%$	$86,\!00\%$	$97{,}69\%$	89,36%
Acurácia: $90,03\%$	Saudável	$92{,}67\%$	$90{,}77\%$	$96{,}52\%$	$91{,}48\%$
InceptionResNetV2-LSTM	COVID-19	$93,\!33\%$	$89,\!23\%$	$97,\!39\%$	$90,\!91\%$
(Configuração 2)	Pneumonia	$87{,}14\%$	88,00%	$93,\!08\%$	$87,\!28\%$
			Continua	na próxir	na página

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 90,02%	Saudável	$90,\!88\%$	$92{,}31\%$	$94,\!82\%$	$91{,}57\%$
POCOVID-Net-2-LSTM	COVID-19	90,00%	90,77%	$93,\!91\%$	$90,\!37\%$
(Configuração 2)	Pneumonia	88,00%	$84,\!00\%$	$97,\!69\%$	$85{,}33\%$
Acurácia: $90,00\%$	Saudável	90,59%	$93,\!85\%$	$93,\!04\%$	$92,\!00\%$
DenseNet201-LSTM	COVID-19	87,78%	90,77%	$90,\!43\%$	89,03%
(Configuração 2)	Pneumonia	$87{,}14\%$	$90,\!00\%$	$98{,}46\%$	$91{,}76\%$
Acurácia: $90{,}00\%$	Saudável	$90{,}91\%$	$89{,}23\%$	$95{,}65\%$	90,00%
ResNet152V2-LSTM	COVID-19	88,18%	$93,\!85\%$	88,70%	90,29%
(Configuração 2)	Pneumonia	$95{,}00\%$	$86,\!00\%$	$99{,}23\%$	$88{,}57\%$
Acurácia: 89,44 $\%$	Saudável	90,00%	$87{,}69\%$	$95{,}65\%$	88,70%
POCOVID-Net-3-LSTM	COVID-19	$91,\!32\%$	$93,\!96\%$	$93,\!91\%$	$92,\!45\%$
(Configuração 2)	Pneumonia	$83,\!33\%$	$80,\!00\%$	$95{,}44\%$	$81{,}08\%$
Acurácia: 88,96%	Saudável	90,00%	$90{,}77\%$	$93,\!91\%$	$90{,}31\%$
MobileNetV2-LSTM	COVID-19	88,33%	87,69%	$93,\!91\%$	88,00%
(Configuração 2)	Pneumonia	$88,\!57\%$	$92,\!00\%$	$93{,}85\%$	$90,\!00\%$
Acurácia: 88,33%	Saudável	88,00%	$86{,}15\%$	$94{,}78\%$	$86{,}96\%$
DenseNet121-LSTM	COVID-19	85,79%	$90,\!88\%$	89,60%	$87,\!91\%$
(Configuração 2)	Pneumonia	$88{,}68\%$	$82,\!00\%$	$96{,}21\%$	$84{,}88\%$
Acurácia: $87{,}31\%$	Saudável	$89{,}03\%$	$87{,}69\%$	$94{,}88\%$	$88{,}16\%$
NasNetLarge-LSTM	COVID-19	83,81%	83,08%	94,82%	82,73%
(Configuração 2)	Pneumonia	$88,\!00\%$	$88,\!00\%$	$93{,}08\%$	$87{,}49\%$
Acurácia: 86,14%	Saudável	$86{,}57\%$	$87{,}69\%$	$91{,}34\%$	$87,\!03\%$
POCOVID-Net-5-LSTM	COVID-19	$88,\!68\%$	83,48%	$93,\!95\%$	85,72%
(Configuração 2)	Pneumonia	$84{,}01\%$	$86,\!00\%$	$90{,}88\%$	$84{,}16\%$
Acurácia: 85,74%	Saudável	87,11%	87,69%	$93,\!91\%$	86,97%
Xception-LSTM	COVID-19	83,64%	88,20%	86,09%	84,78%
(Configuração 2)	Pneumonia	$71,\!52\%$	68,00%	97,77%	$69,\!05\%$
			Continua	na próxir	na página

Tabela 28: continuação da página anterior

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 80,69%	Saudável	$78{,}32\%$	83,08%	$86,\!26\%$	79,75%
VGG16-LSTM	COVID-19	$72,\!65\%$	89,34%	$71,\!34\%$	$78,\!33\%$
(Configuração 2)	Pneumonia	$64{,}17\%$	$58,\!00\%$	$94{,}77\%$	$60,\!40\%$
Acurácia: 73,54 $\%$	Saudável	$82{,}06\%$	$69{,}23\%$	$93,\!12\%$	$71{,}94\%$
VGG19-LSTM	COVID-19	$86,\!62\%$	$74,\!24\%$	91,41%	$72{,}29\%$
(Configuração 2)	Pneumonia	$65{,}71\%$	$44,\!00\%$	$94{,}02\%$	$49{,}86\%$
Acurácia: 70,54 $\%$	Saudável	$69{,}84\%$	$86{,}59\%$	$69{,}38\%$	$74{,}12\%$
POCOVID-Net-1-LSTM	COVID-19	56,79%	80,07%	56,92%	$59{,}53\%$
(Configuração 2)	Pneumonia	$30,\!00\%$	$10,\!00\%$	$97{,}78\%$	$14{,}17\%$
Acurácia: $53,30\%$	Saudável	$64{,}88\%$	$58,\!90\%$	$72{,}59\%$	$52,\!03\%$

Tabela 28: continuação da página anterior

Tabela 29: Resultados da configuração 3 (15 frames).

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
NasNetMobile-LSTM	COVID-19	$91,\!25\%$	$93,\!85\%$	$93,\!91\%$	$92,\!41\%$
(Configuração 3)	Pneumonia	$92{,}50\%$	90,00%	$97{,}69\%$	$91{,}11\%$
Acurácia: 92,22%	Saudável	$93{,}33\%$	$92{,}31\%$	$96{,}52\%$	$92{,}80\%$
InceptionResNetV2-LSTM	COVID-19	$95,\!56\%$	90,77%	98,26%	92,73%
(Configuração 3)	Pneumonia	88,00%	$84,\!00\%$	$97{,}69\%$	$85{,}33\%$
Acurácia: $90,56\%$	Saudável	$89{,}09\%$	$95{,}38\%$	$89{,}57\%$	$91{,}43\%$
Xception-LSTM	COVID-19	90,00%	90,77%	$93,\!91\%$	$90,\!37\%$
(Configuração 3)	Pneumonia	$92{,}50\%$	90,00%	$97{,}69\%$	$91,\!11\%$
Acurácia: $90,00\%$	Saudável	$88{,}57\%$	$89{,}23\%$	$93{,}04\%$	$88{,}89\%$
VGG16-LSTM	COVID-19	88,00%	83,08%	$97,\!39\%$	84,44%
(Configuração 3)	Pneumonia	$89{,}47\%$	$98{,}00\%$	$92{,}31\%$	$92,\!41\%$
Acurácia: $90,00\%$	Saudável	$91{,}67\%$	$90{,}77\%$	$95{,}65\%$	$91{,}20\%$
NasNetLarge-LSTM	COVID-19	91,67%	89,52%	$95,\!65\%$	90,55%
(Configuração 3)	Pneumonia	$85{,}56\%$	88,00%	$94{,}77\%$	$86,\!60\%$
			Continua	a na próxir	na página

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 89,53%	Saudável	$90,\!67\%$	90,77%	$93,\!91\%$	$90,\!63\%$
ResNet152V2-LSTM	COVID-19	$90,\!59\%$	$93,\!85\%$	$93,\!04\%$	$92,\!00\%$
(Configuração 3)	Pneumonia	$86,\!67\%$	$82,\!00\%$	$98{,}46\%$	$83,\!08\%$
Acurácia: 88,89%	Saudável	$87,\!50\%$	$89,\!23\%$	$91{,}30\%$	$88{,}28\%$
DenseNet201-LSTM	COVID-19	90,00%	$92,\!31\%$	$92,\!17\%$	$90,\!81\%$
(Configuração 3)	Pneumonia	$85{,}56\%$	$84,\!00\%$	$97{,}01\%$	$83,\!55\%$
Acurácia: 88,39%	Saudável	89,00%	$88{,}02\%$	$93{,}04\%$	$88{,}35\%$
POCOVID-Net-2-LSTM	COVID-19	$90,\!82\%$	83,08%	$97,\!39\%$	$85,\!26\%$
(Configuração 3)	Pneumonia	$93{,}96\%$	$80,\!00\%$	$97{,}80\%$	$84{,}27\%$
Acurácia: 85,72%	Saudável	$82{,}79\%$	$92{,}31\%$	$82{,}68\%$	$85{,}84\%$
MobileNetV2-LSTM	COVID-19	88,76%	84,62%	$95,\!65\%$	85,77%
(Configuração 3)	Pneumonia	$80,\!00\%$	$76{,}00\%$	$100,\!00\%$	$77{,}89\%$
Acurácia: 85,59%	Saudável	$83,\!42\%$	$93,\!85\%$	$81,\!74\%$	$86{,}74\%$
POCOVID-Net-1-LSTM	COVID-19	86,09%	$86,\!26\%$	86,99%	$84,\!26\%$
(Configuração 3)	Pneumonia	$84,\!85\%$	$84,\!00\%$	$96{,}18\%$	$84,\!05\%$
Acurácia: $84,52\%$	Saudável	$85,\!87\%$	$83{,}19\%$	$93{,}04\%$	$83,\!23\%$
POCOVID-Net-5-LSTM	COVID-19	87,49%	87,80%	$91,\!30\%$	$87,\!37\%$
(Configuração 3)	Pneumonia	$81{,}94\%$	$78{,}00\%$	$93{,}93\%$	$79{,}47\%$
Acurácia: 83,41%	Saudável	80,84%	$83{,}19\%$	$89,\!60\%$	$81,\!74\%$
DenseNet121-LSTM	COVID-19	88,15%	$85,\!23\%$	86,09%	$84{,}62\%$
(Configuração 3)	Pneumonia	$87{,}73\%$	$80,\!00\%$	$96{,}29\%$	$81{,}59\%$
Acurácia: 81,33%	Saudável	$75{,}38\%$	$78,\!57\%$	$88,\!98\%$	$75,\!39\%$
POCOVID-Net-4-LSTM	COVID-19	$79,\!17\%$	$86,\!26\%$	$87,\!86\%$	$82,\!38\%$
(Configuração 3)	Pneumonia	$81{,}34\%$	$94{,}00\%$	$88{,}49\%$	$86,\!44\%$
Acurácia: 81,31%	Saudável	$86{,}90\%$	$66{,}37\%$	$95{,}69\%$	$74{,}69\%$
EfficientNetB0-LSTM	COVID-19	80,27%	77,65%	$85,\!25\%$	77,43%
(Configuração 3)	Pneumonia	71,41%	78,00%	88,02%	74,33%
			Continua	a na próxir	na página

Tabela 29: continuação da página anterior

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 75,42%	Saudável	81,18%	$70{,}99\%$	89,71%	$73,\!41\%$
DenseNet169-LSTM	COVID-19	$77,\!05\%$	65,70%	$90,\!47\%$	$70,\!37\%$
(Configuração 3)	Pneumonia	$65{,}15\%$	$70{,}00\%$	$82{,}59\%$	$66,\!43\%$
Acurácia: 72,15%	Saudável	$75{,}63\%$	$80,\!33\%$	$85{,}59\%$	$77{,}56\%$
POCOVID-Net-3-LSTM	COVID-19	$70,\!15\%$	$72,\!18\%$	80,11%	$70,\!10\%$
(Configuração 3)	Pneumonia	$83,\!28\%$	$66{,}00\%$	$91{,}65\%$	$68{,}66\%$
Acurácia: 70,00%	Saudável	$68{,}11\%$	$71{,}10\%$	$82{,}19\%$	$69{,}33\%$
VGG19-LSTM	COVID-19	$77,\!97\%$	$79,\!95\%$	$74{,}96\%$	$74{,}72\%$
(Configuração 3)	Pneumonia	$81,\!86\%$	$62{,}00\%$	$91{,}77\%$	$62{,}33\%$
Acurácia: $69{,}51\%$	Saudável	$61{,}32\%$	$65{,}49\%$	$86{,}93\%$	$60,\!67\%$

Tabela 29: continuação da página anterior

Tabela 30: Resultados da configuração 4 (20 frames).

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Xception-LSTM	COVID-19	$94,\!44\%$	$100,\!00\%$	$95,\!65\%$	96,77%
(Configuração 4)	Pneumonia	$94{,}55\%$	$96,\!00\%$	$97{,}69\%$	$95{,}24\%$
Acurácia: $95{,}00\%$	Saudável	$97{,}14\%$	$89{,}23\%$	$99{,}13\%$	$92{,}00\%$
DenseNet121-LSTM	COVID-19	$96,\!67\%$	$95{,}38\%$	98,26%	96,00%
(Configuração 4)	Pneumonia	$89{,}61\%$	88,00%	$97,\!00\%$	$88{,}46\%$
Acurácia: 92,82%	Saudável	$91{,}76\%$	$93{,}85\%$	$93{,}91\%$	$92{,}53\%$
InceptionResNetV2-LSTM	COVID-19	$95{,}56\%$	89,23%	$98,\!26\%$	$91,\!93\%$
(Configuração 4)	Pneumonia	89,33%	$94,\!00\%$	$93,\!85\%$	$91,\!20\%$
Acurácia: $91,\!13\%$	Saudável	$90,\!33\%$	90,77%	$94{,}78\%$	$90{,}51\%$
NasNetLarge-LSTM	COVID-19	90,00%	$92,\!31\%$	93,04%	$91,\!03\%$
(Configuração 4)	Pneumonia	$90,\!00\%$	$84,\!00\%$	$98{,}46\%$	85,71%
Acurácia: $91,\!11\%$	Saudável	$92{,}50\%$	$95{,}38\%$	$94{,}78\%$	$93{,}79\%$
ResNet152V2-LSTM	COVID-19	88,75%	90,77%	$92,\!17\%$	$89,\!66\%$
(Configuração 4)	Pneumonia	$92,\!00\%$	86,00%	$98{,}46\%$	88,00%
			Continua	na próxir	na página

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 90,56%	Saudável	92,00%	$93,\!85\%$	$94{,}78\%$	$92{,}86\%$
POCOVID-Net-4-LSTM	COVID-19	$92,\!00\%$	$87,\!69\%$	96,52%	$89,\!63\%$
(Configuração 4)	Pneumonia	$88,\!57\%$	$92,\!00\%$	$93,\!85\%$	$90,\!00\%$
Acurácia: $90,02\%$	Saudável	$90,\!33\%$	90,77%	$94{,}78\%$	$90{,}51\%$
NasNetMobile-LSTM	COVID-19	90,00%	$98,\!46\%$	89,57%	$92,\!97\%$
(Configuração 4)	Pneumonia	$92{,}00\%$	$86,\!00\%$	$98{,}46\%$	$88,\!00\%$
Acurácia: $90{,}00\%$	Saudável	$88{,}57\%$	$84{,}62\%$	$96{,}52\%$	$86,\!00\%$
VGG16-LSTM	COVID-19	$90,\!91\%$	89,23%	$95,\!65\%$	90,00%
(Configuração 4)	Pneumonia	$86{,}67\%$	$82,\!00\%$	$98{,}46\%$	$83,\!08\%$
Acurácia: $90{,}00\%$	Saudável	90,00%	$96{,}92\%$	$90{,}43\%$	$92{,}57\%$
DenseNet169-LSTM	COVID-19	88,00%	$86,\!15\%$	$94,\!78\%$	$86,\!96\%$
(Configuração 4)	Pneumonia	$90,\!00\%$	$86,\!00\%$	$97{,}69\%$	$87,\!50\%$
Acurácia: 88,89%	Saudável	89,00%	$93{,}85\%$	$90{,}43\%$	$90{,}91\%$
POCOVID-Net-2-LSTM	COVID-19	90,00%	$86,\!15\%$	$96{,}52\%$	$87,\!26\%$
(Configuração 4)	Pneumonia	$88,\!18\%$	$88,\!00\%$	$95{,}46\%$	$87{,}99\%$
Acurácia: 88,39%	Saudável	$88{,}05\%$	$90{,}77\%$	$90{,}47\%$	$88{,}91\%$
EfficientNetB0-LSTM	COVID-19	88,80%	90,99%	$92,\!21\%$	$89{,}52\%$
(Configuração 4)	Pneumonia	$90{,}65\%$	$88,\!00\%$	$96{,}95\%$	$88,\!81\%$
Acurácia: 88,38%	Saudável	$86,\!90\%$	$86{,}15\%$	$93{,}08\%$	$86{,}46\%$
POCOVID-Net-1-LSTM	COVID-19	89,33%	$87{,}98\%$	93,04%	$88,\!55\%$
(Configuração 4)	Pneumonia	$80{,}38\%$	$80,\!00\%$	$94{,}01\%$	$79{,}84\%$
Acurácia: 86,77%	Saudável	$88{,}67\%$	$90{,}77\%$	$93{,}04\%$	$89,\!62\%$
DenseNet201-LSTM	COVID-19	$87,\!32\%$	90,77%	$91,\!30\%$	$88{,}91\%$
(Configuração 4)	Pneumonia	$88{,}18\%$	$82{,}00\%$	$97{,}69\%$	83,71%
Acurácia: 86,68%	Saudável	86,17%	$86,\!15\%$	$90{,}43\%$	$85,\!92\%$
MobileNetV2-LSTM	COVID-19	$85,\!90\%$	87,80%	$91,\!38\%$	86,72%
(Configuração 4)	Pneumonia	$75{,}68\%$	$68,\!00\%$	$97{,}72\%$	71,07%
			Continua	na próxir	na página

Tabela 30: continuação da página anterior

Classificador	Classe	Prec.	Sens.	Espec.	F1-Score
Acurácia: 82,87%	Saudável	$80,\!37\%$	$89,\!34\%$	84,38%	84,12%
POCOVID-Net-5-LSTM	COVID-19	82,13%	$83,\!59\%$	88,73%	82,75%
(Configuração 4)	Pneumonia	$74{,}97\%$	$80,\!00\%$	$90,\!39\%$	$76{,}75\%$
Acurácia: 81,96%	Saudável	$90,\!00\%$	$81{,}76\%$	$93{,}91\%$	$85{,}57\%$
VGG19-LSTM	COVID-19	$63,\!98\%$	94,24%	61,92%	$74,\!84\%$
(Configuração 4)	Pneumonia	$67{,}18\%$	$64,\!00\%$	$94{,}89\%$	$64,\!11\%$
Acurácia: 67,19%	Saudável	80,99%	$42{,}53\%$	$92{,}38\%$	$51,\!17\%$
POCOVID-Net-3-LSTM	COVID-19	$60,\!59\%$	$73{,}61\%$	$72,\!50\%$	65,73%
(Configuração 4)	Pneumonia	$80,\!67\%$	$32,\!00\%$	$94{,}85\%$	$41{,}56\%$
Acurácia: $62,06\%$	Saudável	$63,\!37\%$	72,75%	$73,\!71\%$	$66{,}62\%$

Tabela 30: continuação da página anterior