

**NIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE ELÉTRICA**

**CLASSIFICAÇÃO AUTOMÁTICA PARA O DIAGNOSTICO DE
LESÕES DE PELE USANDO DEEP CONVOLUTIONAL
NEURAL NETWORK COM MODELOS
DE TRANSFER LEARNING**

por

Derick Abreu Montagna

Itajaí (SC), dezembro de 2021

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA ELÉTRICA**

**CLASSIFICAÇÃO AUTOMÁTICA PARA O DIAGNOSTICO DE
LESÕES DE PELE USANDO DEEP CONVOLUTIONAL
NEURAL NETWORK COM MODELOS
DE TRANSFER LEARNING**

Área de Inteligência Artificial.

por

Derick Abreu Montagna

Relatório apresentado à Banca Examinadora do Trabalho Técnico-científico de Conclusão de Curso de Engenharia Elétrica para análise e aprovação.
Orientador(a): Rudimar Luis Scaranto Dazzi, Dr.

Itajaí (SC), dezembro de 2021

*Dedico a mim mesmo e todas as pessoas que cruzaram a minha vida até a entrega
desse trabalho em 11 de novembro de 2021.*

AGRADECIMENTOS

Os meus agradecimentos serão por ordem dos acontecimentos de toda a minha trajetória da minha graduação. Iniciando no primeiro período, com o meu primeiro trabalho extracurricular, a construção de uma câmera anecoica junto ao professor Carlos Alberto Mortiz Jr, agradeço a ele pela oportunidade.

Assim como, nesse mesmo período, pude conhecer uma professora que mudaria a minha visão do mundo da pesquisa e a minha visão de mundo, a professora Albertina Xavier da Rosa Corrêa. De maneira, que durante os próximos dois anos e meios pude focar em pesquisas sobre eletrofloculação, do segundo ao sexto período. Uma vez que, pude consistar uma bolsa de pesquisa do FUMDES, assim como, pude participar de diversos eventos de pesquisa e inovação: Prêmio Univali de Inovação e a FEBIC em Jaraguá do Sul. Agradeço imensamente a professora Albertina e ao professor Renan C. Testolin por ter dado a oportunidade inicial para um calouro iniciar no ramo da pesquisa.

Outro professor, que me impactou durante a minha graduação, que me fez amar cálculo e eletromagnetismo, agradeço ao professor Marcelo de Paoli. Assim como, o professor Paoli, tem um dom em ser um ótimo professor, outro que me impactou e ministrava suas aulas com maestria, e agradeço muito por fazer parte do meu desenvolvimento foi o professor Raimundo Teive.

Uma disciplina que me afetou de diversas maneiras, tanto como pessoa, quanto na minha forma de ver o mundo, foi estatística do sexto período, ministrada pela professor Rodrigo Sant'ana. Visto que, pude obter muitos ensinamentos sobre a área que sigo atualmente, ciência de dados, assim como, puder entender como a ciência deve ser perante a sociedade. Nessa mesma disciplina, pude conhecer alguém muito especial para mim, a minha namorada, Andressa Leite da Silva. Amo muito você.

Agradeço também, ao professor Rudimar Dazzi, meu orientador nesse trabalho e na minha pesquisa do PIBITI, uma ótima pessoa que sempre me deu diversos conselhos sobre as direções corretas para as minhas pesquisas. Como também, ao professor Wemerson Parreira, pelas oportunidades e ensinamentos acerca de publicações. E, ao professor Rafael Ballotin Martins por me introduzir ao mundo da ciência dos dados e me fazer escolher em definitivo o caminho que eu gostaria de trilhar.

Agradeço a todos aos meus colegas de classe, os sobreviventes do curso, Edmiel, Leonardo, Bruno, Daniel e Eduarda. E aos demais professores que ministraram disciplinas ao longo do curso.

Agradeço também, ao Guilherme Klimke, por acreditar no meu potencial e me ajudar na minha primeira empreitada no mundo corporativo. Assim como, ao Julierme Arrais e toda equipe de TI da Cantu Store, em especial, a equipe de dados: Fejão (Matheus), Cinthia, Alberto e Leonardo Neto.

Agradeço aos meus familiares, meu pai, Paulo Roberto Montagna, por sempre fazer de tudo por mim e meus irmãos. Minha mãe, Rosangela Quadros de Abreu por sempre me apoiar. Ao meu irmão Ramsés Prestes Montagna pelos diversos conselhos e fornece uma central de energia por meio do Café Paris. Como também, meu tio e tia Luiz e Mimo; e minhas primas Fran e Le.

Por fim, a mim mesmo, por nunca desistir e sempre buscar evoluir da melhor maneira. Ajudando diversas pessoas no processo e crescendo muito como pessoa e ser humano. Espero, que consiga trilhar o caminho de seu mestrado, doutorado e ir morar fora do país.

Me despeço desses agradecimentos e desse trabalho em 11/11/2021 às 20:46 em uma quinta-feira, em um dia nublado.

Falamos não apenas para dizer aos outros o que pensamos, mas para dizer a nós mesmos o que pensamos.
- Oliver Sacks

RESUMO

Montagna, Derick Abreu. Classificação automática para o diagnóstico de lesões de pele usando deep convolutional neural network com modelos de transfer learning. Itajaí, 2021. 99 f. Trabalho Técnico-científico de Conclusão de Curso (Graduação em Engenharia Elétrica) – Escola do Mar, Ciência e tecnologia da Universidade do Vale do Itajaí, Itajaí, 2021.

No Brasil, o câncer de pele vem se tornando a neoplasia mais frequente entre os pacientes. Se detectado no início, há grandes chances de cura. O diagnóstico de lesões de pele é naturalmente um desafio e em regiões que não contam com dermatologistas, o problema é ainda maior. Um estudo demonstrou, que os dermatologistas obtiveram 93% e as outras especializações 52% para a capacidade de indicar a morbidade das lesões, o que evidencia a importância de um profissional capacitado para a realização dessas análises. Porém, o Brasil sofre com uma grande desigualdade na distribuição da população médica entre regiões, estados, capitais e municípios do interior. Portanto, a falta de profissionais da saúde habilitados para realizar tais procedimentos em certas regiões, acaba dificultando o processo de diagnóstico precoce, elevando a taxa de mortalidade no Brasil. De tal maneira, ferramentas de apoio a decisão tornam-se relevantes tecnologias para contribuir com os profissionais e com o sistema de saúde, como suporte para identificar o tipo de lesão de pele e direcionar o profissional da saúde e/ou paciente que não detém de conhecimentos sobre a área de dermatologia. Por isso, esse trabalho tem como objetivo, desenvolver um modelo capaz de diagnosticar os tipos de lesões de pele presentes no conjunto de dados HAM1000 por meio da classificação das imagens dermoscópicas, utilizando métodos e técnicas de processamento de imagens e redes de aprendizado profundo usados para a tarefa de classificação. Portanto, para a implementação do modelo foi utilizado um fluxo dividido em quatro etapas: carga do HAM10000, pré-processamento único (aplicação de *Shades of Gray*, validação cruzada de k-partes), pré-processamento regular (camadas de redimensionamento, aumento de dados e reescalonamento), e por fim, classificação (modelo base entre os doze modelos pré-treinados seguido de uma camada do topo constituído de uma *global average pooling* e uma camada de *Dropout*). A avaliação do modelo foi realizada em três momentos: treinamento (utilizando o fluxo descrito anteriormente), validação (removendo apenas a validação cruzada de k-partes do fluxo descrito anteriormente e a realização do treino do modelo com a base total do HAM10000) e teste (o mesmo da validação). Os momentos de treinamento e validação foram avaliados por meio das métricas de avaliação utilizadas as tabelas-verdades: matriz de confusão, curva auc-roc, curva precisão-revocação e a acurácia balanceada; e o momento de teste por meio da submissão no site do ISIC resultando a métrica de avaliação acurácia balanceada. Por fim, foi realizado a seleção dos seis melhores modelos com as maiores acurárias balanceada para a realização do *ensemble*. O modelo *ensemble* alcançou um valor final de 74,50% de acurácia balanceada. Sobretudo, a família de modelos pré-treinado DenseNet demonstrou ótimo resultados para a classificação de lesões de pele. Como contribuições, o fluxograma dos processos de treinamento, validação e teste para a classificação de lesões de pele. Assim como, uma forma eficiente de lidar com o desbalanceamento do conjunto de dados, encontrar rapidamente uma ótima taxa de aprendizado para o treinamento e a demonstração da importância da validação cruzada.

Palavras-chave: HAM10000. Câncer de pele. Deep Learning.

ABSTRACT

In Brazil, skin cancer has become the most frequent neoplasm among patients. If detected early, chances of cure are high. The diagnosis of skin lesions is naturally a challenge and in regions that do not have dermatologists, the problem is even greater. A study showed that dermatologists obtained 93% and other specializations 52% for the ability to indicate the morbidity of the lesions, which highlights the importance of a trained professional to carry out these analyses. However, Brazil suffers from a great inequality in the distribution of the medical population between regions, states, capitals and municipalities in the interior. Therefore, the lack of qualified health professionals to perform such procedures in certain regions ends up hindering the process of early diagnosis, increasing the mortality rate in Brazil. In this way, decision support tools become relevant technologies to contribute to professionals and the health system, as support to identify the type of skin lesion and direct health professionals and/or patients who do not have knowledge in the field of dermatology. Therefore, this work aims to develop a model capable of diagnosing the types of skin lesions present in the HAM1000 dataset through the classification of dermoscopic images, using methods and techniques of image processing and deep learning networks used for the classification task. Therefore, a flow divided into four steps was used to implement the model: loading of the HAM10000, single pre-processing (Shades of Gray application, k-parts cross validation), regular pre-processing (resizing layers, data increase and rescheduling), and finally, classification (base model among the twelve pre-trained models followed by a top layer consisting of a global average pooling and a Dropout layer). The evaluation of the model was performed in three moments: training (using the flow described above), validation (removing only the cross-validation of k-parts of the flow described above and the performance of training the model with the total base of the HAM1000) and testing (same as validation). The training and validation moments were evaluated using the evaluation metrics used in the truth tables: confusion matrix, auc-roc curve, precision-recall curve and balanced accuracy; and the time of testing through the submission on the ISIC website resulting in a balanced accuracy assessment metric. Finally, the selection of the six best models with the highest balanced accuracies was performed to carry out the ensemble. The ensemble model reached a final value of 74.50% of balanced accuracy. Above all, the family of pre-trained DenseNet models showed excellent results for the classification of skin lesions. As contributions, the flowchart of the training, validation and testing processes for the classification of skin lesions. As well as an efficient way to deal with dataset imbalance, quickly find an optimal learning rate for training and demonstrate the importance of cross-validation.

Keywords: HAM10000. Skin Lesions. Deep Learning.

LISTA DE FIGURAS

Figura 1. Imagem 2x2px em escala cinza.....	26
Figura 2. Imagem 2x2px em escala RGB	27
Figura 3. Dermatoscópio de polarização sem fio DE350, com ampliação óptica de 15x e 45x digital.....	29
Figura 4. <i>Exemplos de uma imagem antes e depois da aplicação do algoritmo Shades of Gray com a distância de Minkowski Normalizada igual a 6</i>	34
Figura 5. Exemplos de aplicações de técnicas de aumento de dados em uma imagem DF	35
Figura 6. Anatomia do neurônio artificial	38
Figura 7. Grafo arquitetural de um perceptron de múltiplas camadas com duas camadas ocultas	39
Figura 8. Representação gráfica da operação de correlação cruzada para uma entrada 2D....	42
Figura 9. Calculando os valores de max pooling.....	43
Figura 10. Exemplo de uma arquitetura típica CNN	44
Figura 11. Função de ativação ReLU	46
Figura 12. Retro propagação	48
Figura 13. Fluxograma de aprendizagem de uma rede neural.....	49
Figura 14. Visualizações dos resultados dos filtros em diferentes níveis para duas arquiteturas de CNNs diferentes.....	50
Figura 15. Demonstração do k-fold cross validation.....	53
Figura 16. Execução do LR. O valor encontra como melhor taxa de aprendizado foi de 1E-3, para um modelo de InceptionV3	54
Figura 17. Ponto de sela	56
Figura 18. Abordagem de execução do algoritmo do CLR	56
Figura 19. Fluxograma de desenvolvimento geral do modelo	60
Figura 20. Processos realizados durante a implementação.....	61
Figura 21. Fluxograma de desenvolvimento do modelo geral detalhado.....	63
Figura 22. Fluxograma do pré-processamento utilizado no treinamento do modelo	66
Figura 23. Exemplo da aplicação da virada aleatória horizontal e vertical	67
Figura 24. Exemplo da aplicação da rotação aleatória	68
Figura 25. Exemplo da aplicação do contraste aleatório.	68
Figura 26. Primeiro momento, fluxograma de classificação: Congelamento do modelo base e a realização do treino somente das camadas do topo	70
Figura 27. Exemplo de análise pós-treinamento do primeiro momento do modelo.....	71
Figura 28. Segundo momento, fluxograma de classificação: afinamento do modelo total.....	72
Figura 29. Exemplo de análise pos-treinamento do segundo momento do modelo	73
Figura 30. Fluxograma resumindo o processo de treinamento.....	74
Figura 31. Fluxograma de validação e teste com um subconjunto fornecido pelo ISIC 2018 Challenge	75
Figura 32. Método de aprendizado <i>ensemble</i> : Média das previsões	76

Figura 33. Representação em dispersão dos resultados obtidos com o treinamento com KFCV	77
Figura 34. Formas do afinamento.....	78
Figura 35. Comparativo entre as acurárias categoria e perdas de treinamento e validação sobre a época do processo abordado na seção 3.3.....	80
Figura 36. Matrizes de confusão de oito modelos	81
Figura 37. Curvas ROC-AUC dos seis modelos com maioria acurácia balanceada no conjunto de validação	83
Figura 38. Curva de Precisão-Revocação dos seis modelos com maioria acurácia balanceada no conjunto de validação	84
Figura 39. Um bloco denso com 5 camadas e taxa de crescimento 4	101
Figura 40. Uma DenseNet profunda com três blocos densos.....	102
Figura 41. Aprendizagem residual: um bloco de construção	104
Figura 42. Convolução Separável em Profundidade Original	106
Figura 43. A convolução separável em profundidade modificada	107
Figura 44. A arquitetura da Xception para o ImageNet challenge	107
Figura 45. Módulo Inception Simples	108
Figura 46. Módulo Inception com redução de dimensões.....	109
Figura 47. Tipos de módulos Inception	110
Figura 48. Módulos de iniciação A, B, C em um Inception-ResNet-V2 Observe como a camada de <i>pooling</i> foi substituída pela conexão residual e a convolução 1x1 adicional antes da adição.	110
Figura 49. Bloco Residual Inception A (redução de tamanho de 35×35 a 17×17), Bloco Residual Inception B (redução de tamanho de 17×17 a 8×8) e Bloco Residual Inception C (tamanho de 8×8).....	111
Figura 50. As ativações são escaladas por uma constante para evitar que a rede morra.....	111
Figura 51. A arquitetura da Inception-ResNet para o ImageNet challenge.....	112
Figura 52. Comparação de diferentes métodos de escalonamento. (a) é um exemplo de uma rede padrão; (b) - (d) são escalas convencionais que apenas aumentam uma dimensão da rede largura, profundidade ou resolução. (e) método de escala composta proposto que escala uniformemente todas as três dimensões com uma razão fixa.....	113
Figura 53. Matriz de confusão genérica	116
Figura 54. Demonstração do Trade-Off Recall-Precisão	118
Figura 55. Exemplo da curva de precisão-recall	118
Figura 56. Curva AUC-ROC	120
Figura 57 - Trade-Off Viés-Variância	122
Figura 58. Principais fontes encontradas na busca no texto completo das publicações por meio da string de busca	127
Figura 59. Principais fontes encontradas na busca no título e resumo das publicações por meio da string de busca	128

LISTA DE QUADROS

Quadro 1.	Comparativo entre as sete categorias presentes no HAM10000	32
Quadro 2.	Comparativo entre os modelos pré-treinados.....	51
Quadro 3.	Comparaçao entre os trabalhos selecionados	59
Quadro 4.	Bibliotecas e versões utilizadas na data desse trabalho.....	62
Quadro 5.	Especificações do ambiente Kaggle e Colab Pro	62
Quadro 6.	Distribuição dos dados	63
Quadro 7.	Pesos associados a cada categoria no momento do treinamento do modelo 64	
Quadro 8.	Distribuição das variantes das imagens.....	65
Quadro 9.	Resultados do treinamento com KFCV.....	77
Quadro 10.	Resultado do subconjunto de validação	79
Quadro 11.	Resultados do subconjunto de teste.....	86
Quadro 12.	Comparaçao com os trabalhos relacionados	87
Quadro 13.	Arquiteturas DenseNet para ImageNet. A taxa de crescimento para todas as redes é $k = 32$. Observe que cada camada "conv" mostrada abaixo corresponde a sequênci ReLU-Conv 103	
Quadro 14.	Arquiteturas ResNet para ImageNet. Os blocos de construção são mostrados entre parênteses, com os números de blocos empilhados. A redução da resolução é realizada por conv3 1, conv4 1 e conv5 1 com uma passada de 2.....	105
Quadro 15.	A arquitetura InceptionV2/V3 proposta para o para o ImageNet challenge. O tamanho de saída de cada módulo é o tamanho de entrada do próximo.	109
Quadro 16.	Arquitetura base EfficientNet-B0 para a ImageNet Challenge.....	114
Quadro 17.	Métricas que formam a curva de operação do receptor (ROC - Receiver operating Characteristic).....	120
Quadro 18. tratamentos	Demonstração dos três tipos de diagnostico que um modelo pode ter e seus 121	

LISTA DE ABREVIATURAS E SIGLAS

ACS	American Cancer Society
AKIEC	Ceratoses Actínicas e Carcinoma Espinocelular
API	Application Programming Interface
AUC	Area abaixo da curva
BCC	Carcinoma Basocelular
BKL	Ceratose benigna (ceratose seborreica/ lentigo/ liquen plano)
CA	Ceratose Actínia
CEC	Carcinoma Espinocelular
CFM	Conselho Federal de Medicina
CNN	Rede Neural Convolucional
CREMESP	Conselho Regional de Medicina do Estado de São Paulo
CV	Validação cruzada
DCNN	Redes neurais convolucionais profundas
DA	Aumento de dados
DF	Dermatofibroma
DL	Aprendizado Profundo
FA	Função de ativação
FP	Falso Positivo
FN	Falso Negativo
HAM10000	Human Against Machine with 10000 training images
HAM	Human Against Machine with 10000 training images
INCA	Instituto Nacional De Câncer José Alencar Gomes Da Silva
ISIC	The International Skin Imaging Collaboration
IA	Inteligência Artificial
IoT	Internet of Things
IoMT	Internet of Medical Things
ISIC	The International Skin Imaging Collaboration
KFCV	Validação cruzada k-partes
LR	Learning Rate Finder
MEL	Melanoma
ML	Aprendizado de máquina
NV	Nevo Melanocítico
RGB	Red-Green-Blue
ReLU	Rectified Linear Unit
RNA	Rede Neural Artificial
SBCD	Sociedade Brasileira de Cirurgia Dermatológica
SVM	Support Vector Machine
TL	Aprendizado por transferência
UV	Raios Ultravioletas
VASC	Lesão Vascular
VP	Verdadeiros Positivo
VPN	Valor preditivo Negativo
VN	Verdadeiros Negativo

LISTA DE SÍMBOLOS

p	Intensidade de cinza
s	Coordenada espacial vertical
d	Coordenada espacial horizontal
T	Transposta
e	Iluminante desconhecido
e^*	Iluminante branco ideal
I	Imagen
q	pixel
q	Coordenada pixel vertical
v	Coordenada pixel horizontal
k	Constante de normalização
n	Grau da Norma de Minkowski
D_p	Distância de Minkowski
c	Coeficientes para o iluminante
x_i	Atributos do sinal
w	Pesos sinápticos
u_k	Potencial de ativação
b_k	bias
y_k	Um neurônio artificial
f	Função de entrada
g	Filtro
K	Filtro bidimensional
S	Mapa de características
f_a	Função de ativação
l_r	Valor máximo de taxa de aprendizado
n	Número de iterações
t_r	Fator de aprendizado
L	Camadas
h	Função
k	Número de classes
VP	Verdadeiro Positivo
VN	Verdadeiro Negativo
FP	Falso Positivo
FN	Falso Negativo
e	Euler
t	Fator de aprendizado
J_{cce}	Função de custo <i>Categorical cross-entropy</i>
$=$	Igualdade
$<$	Menor
\times	Multiplicação
$/$	Divisão
Σ	Somatório
$ $	Módulo
$\sqrt{}$	Raiz
$*$	Convolução
$+$	Soma

SUMÁRIO

1 INTRODUÇÃO	17
1.1 Problematização.....	18
1.1.1 Solução proposta	20
1.1.2 Delimitação de Escopo	20
1.1.3 Justificativa.....	21
1.2 Objetivos.....	22
1.2.1 Objetivo Geral	22
1.2.2 Objetivos Específicos	22
1.3 Metodologia.....	23
1.3.1 Metodologia de Pesquisa	23
1.3.2 Procedimentos Metodológicos	23
1.4 Estrutura do Trabalho	24
2 FUNDAMENTAÇÃO TEÓRICA.....	25
2.1 Imagem digital	25
2.1.1 Representação	25
2.1.1.1 Escala cinza	26
2.1.1.2 Escala colorida.....	26
2.1.2 Imagens médicas.....	27
2.1.2.1 Imagens dermatoscópicas	28
2.2 Dados	29
2.2.1 Dataset HAM10000	29
2.2.1.1 População amostral.....	30
2.2.2 Categorias	31
2.3 Pré-processamento.....	32
2.3.1 Constância de Cor.....	32
2.3.2 Aumento de dados	35
2.3.3 Redimensionamento e Reescalonamento	36
2.4 Classificação	36
2.4.1 Aprendizado Profundo.....	37
2.4.2 Neurônio artificial.....	37
2.4.3 Rede neural artificial	39
2.4.4 Rede neural convolucional	40
2.4.4.1 Camada de Convolução	40
2.4.4.2 Camada de Pooling.....	42
2.4.4.3 Arquitetura típica da CNN.....	44
2.4.5 Funções e otimizadores	45
2.4.5.1 Função de ativação	45

2.4.5.2 Função de custo	47
2.4.5.3 Otimizadores.....	48
2.4.6 Transfer Learning	50
2.4.7 Métodos e algoritmos	53
2.4.7.1 Validação cruzada de k-partes	53
2.4.7.2 Learning Rate Finder.....	54
2.4.7.3 Cyclical Learning Rate	55
2.5 Trabalhos similares	56
2.5.1 Revisão Sistemática da Literatura	56
2.5.2 Trabalhos selecionados.....	57
 3 DESENVOLVIMENTO.....	 60
3.1 Implementação.....	61
3.1.1 Ambiente	61
3.1.1.1 Linguagem de programação e bibliotecas	61
3.1.1.2 Hardware	62
3.2 Treinamento	63
3.2.1 Base de dados: abordagens com o desbalanceamento das classes	63
3.2.2 Pré-processamento.....	65
3.2.3 Classificação	69
3.2.4 Resumo do treinamento	74
3.3 Validação e teste	74
3.4 Ensemble	75
 4 RESULTADOS E DISCUSSÕES	 77
4.1 Comparação com os trabalhos relacionados.....	87
 5 CONSIDERAÇÕES FINAIS.....	 89
 REFERÊNCIAS	 91
 APÊNDICE A. DETALHAMENTO DAS ARQUITETURAS DE REDES NEURAIS CONVOLUCIONAIS PROFUNDAS	 101
A.1. DenseNet	101
A.2. ResNet-V2.....	103
A.3. Xception	105
A.4. InceptionV3	108
A.5. InceptionResnetV2.....	110
A.6. EfficientNet.....	112
 APÊNDICE B. METRICAS DE AVALIAÇÃO	 115

B.1. Matriz de confusão.....	115
B.2. Acurácia	116
B.2.1. Acurácia balanceada	117
B.3. Recall (Sensibilidade) e Precisão (Valor preditivo positivo).....	117
B.3.1. Curva de precisão-recall	118
B.4. F1-Score	119
B.5. Valor preditivo negativo (VPN).....	119
B.6. Especificidade.....	119
B.7. Curva AUC-ROC.....	119
B.8. Análise dos modelos	121
APÊNDICE C. REVISÃO SISTEMATICA	124
C.1. Questões	124
C.2. Repositório bibliográfico.....	125
C.3. Critérios de elegibilidade	125
C.4. A string de busca.....	126
C.5. Resultados.....	127

1 INTRODUÇÃO

As lesões de pele são frequentes e podem ser sinal de doenças graves para o ser humano, como por exemplo, o câncer de pele. O câncer de pele é a neoplasia mais frequente no Brasil. Os principais fatores de risco para o câncer de pele são a exposição prolongada ao sol (raios ultravioletas - UV), principalmente na infância e adolescência, exposição a câmeras de bronzeamento artificial e história familiar de câncer de pele (ACS, 2017; INCA, 2019).

Se detectado no início, há grandes chances de cura, mesmo nas manifestações mais agressivas. Para diagnosticar o câncer de pele, é preciso efetuar a biópsia da lesão. Pode-se também utilizar métodos não invasivos complementares, como a dermatoscopia manual, ou dermatoscopia digital. A dermatoscopia é uma técnica não-invasiva, no qual o dermatologista usa um aparelho chamado dermatoscopia para fotografar a região da lesão, aumentando-a de 10 a 70 vezes (SBCD, 2020).

O diagnóstico de lesões de pele é naturalmente um desafio e em regiões que não contam com dermatologistas, o problema é ainda maior. Uma vez que, nessas regiões há uma menor taxa de assertividade nos diagnósticos das lesões de pele. Tal qual, segundo Federman (1999), que realizou um estudo sobre a capacidade de indicar a morbidade da lesão, os dermatologistas obtiveram 93% e as outras especializações 52%, o que evidencia a importância de um dermatologista para realizar essas análises.

Compreendendo a importância do diagnóstico das lesões de pele em seu estágio precoce, métodos de processamento digital de imagens e algoritmos de Aprendizado Profundo (do inglês, *Deep Learning*, DL) estão sendo aplicados com a utilização das imagens dermatoscópicas. Para aperfeiçoar os métodos de diagnóstico existentes e auxiliar os profissionais da saúde (dermatologistas ou não), novos modelos de diagnóstico automatizados estão sendo propostos.

Portanto, a proposta desse trabalho é desenvolver um modelo que poderá contribuir com o avanço na área de diagnóstico de lesões de pele com a utilização de inteligência artificial (IA). Com o avanço dessa área, esses modelos poderão auxiliar os profissionais da saúde no processo de avaliação do tipo de lesão de pele, possibilitando maior acurácia nos diagnósticos das lesões. Com um diagnóstico rápido e assertivo, o tratamento pode ser agilizado, propiciando um aumento na recuperação do paciente.

Com a finalização do trabalho algumas contribuições são apresentadas, conforme segue:

- Metodologia para lidar com o desbalanceamento dos conjuntos de dados de lesões de pele;
- Demonstração da eficiência da utilização de aprendizagem por transferência (*do inglês, Transfer Learning, TL*) para o diagnóstico de lesões de pele por meio da extração das características e a sua classificação com arquiteturas de redes neurais convolucionais profundas (*do inglês, deep convolutional neural network*); e
- A difusão da tecnologia deste trabalho para a redução significativa de hospitalizações e tratamentos desnecessários devido a diagnósticos errôneos. Como também, leva a uma diminuição dos custos diretos e indiretos relacionados ao diagnóstico de lesões de pele.

1.1 PROBLEMATIZAÇÃO

Mundialmente, o câncer está entre as quatro principais causas de morte prematura (antes dos 70 anos de idade) na maioria dos países. Ao passo que, é o principal problema de saúde pública no mundo. Por conseguinte, vem ocorrendo um aumento no mundo em casos de mortalidade, em parte pelo crescimento populacional e o seu envelhecimento, como também por questões de desenvolvimento econômico e na mudança de desenvolvimento e na prevalência de fatores de risco de câncer (BRAY et al., 2018).

As lesões de pele podem ser classificadas em dois tipos: melanoma e não-melanoma. O melanoma é a lesão de pele mais perigosa e a mais provável de se espalhar sobre outras partes do corpo, se não for detectada, controlada e tratada precocemente. O melanoma é produzido pela reprodução anormal dos melanócitos (células que dão a cor da pele), que começam a crescer rapidamente e fora de controle levando a metástase (ACS, 2017). De todos as neoplasias malignas diagnosticadas no mundo, o câncer de pele não melanoma é o tipo mais frequente em ambos os sexos (BRAY et al. 2018; FERLAY et al., 2018).

O câncer de pele melanoma, foi estimado para cada ano do triênio 2020-2022, em 4.200 em homens e de 4.250 em mulheres em números de novos casos no Brasil. Esses valores correspondem a um risco estimado de 4,03 casos novos a cada 100 mil homens e 3,94 para cada

100 mil mulheres. Na Região Sul, o câncer de pele melanoma é mais incidente quando comparado com as demais Regiões, para ambos os sexos (INCA, 2019).

Já para o câncer de pele não melanoma, dentro do mesmo período, em homens é mais incidente nas Regiões Sul, Centro-Oeste e Sudeste, com um risco estimado de 123,67/100 mil, 89,68/100 mil e 85,55/100 mil, respectivamente. Nas Regiões Nordeste e Norte, ocupa a segunda posição, com um risco estimado de 65,59/100 mil e 21,28/100 mil, respectivamente. No que diz respeito às mulheres, o câncer de pele não melanoma é mais incidente em todas as Regiões brasileiras, com um risco estimado de 125,13/100 mil (Centro-Oeste), 100,85/100 mil (Sudeste), 98,49/100 mil (Sul), 63,02/100 mil (Nordeste) e 39,24/100 mil no Norte (INCA, 2019).

A detecção precoce do câncer é uma estratégia para encontrar um tumor numa fase inicial e, assim, possibilitar maior chance de tratamento. Quanto mais precoce for sua identificação, melhores serão os resultados do tratamento (INCA, 2019). Porém, para a realização dos diagnósticos é necessário profissionais de saúde habilitados para a realização dos procedimentos de diagnóstico das lesões de pele. No entanto, o Brasil sofre com uma grande desigualdade na distribuição da população médica entre regiões, estados, capitais e municípios do interior (CFM e CREMESP, 2020).

De acordo com o estudo de projeção Concentração de Médicos no Brasil em 2020, realizado pelo Conselho Federal de Medicina (CFM) e Conselho Regional de Medicina do Estado de São Paulo (CREMESP), o Brasil atingirá em 2020 a razão de 2,20 médicos por 1.000 habitantes. Dessa forma, sem uma política adequada para distribuição de médicos, o governo federal irá acentuar as desigualdades no acesso à assistência médica a favor de regiões e municípios que já contam com uma alta concentração de profissionais (CFM e CREMESP, 2020).

Para explicitar, o estado do Maranhão mantém a menor razão entre as unidades federativas, com 0,87 médico por mil habitantes, seguido pelo Pará, com razão de 0,97 (DMB, 2018). Segundo Schmitt et al. (2014), que realizaram um estudo investigativo em 2013, sobre a distribuição geográfica dos dermatologistas nos municípios brasileiros, constatou que apenas 504 dos 5.565 municípios brasileiros possuíam especialistas em dermatologia, ou seja, 9,1%.

Ademais, ao fato da desigualdade dos profissionais de saúde, um estudo realizado no ano de 2014 no Estado de Minas gerais, no município de Jequitinhonha, a especialidade Dermatologia foi a segunda mais acionada pelos médicos de atenção primária, totalizando 17,78%

dos encaminhamentos e um tempo de espera média superior a 108 dias (VIEIRA; LIMA; GAZ-ZINELLI, 2015). Levando a uma menor agilidade no tratamento e recuperação do paciente. Podendo causar o óbito do paciente, devido a perca do período precoce da lesão.

Neste contexto, ferramentas de apoio a decisão tornam-se relevantes tecnologias para contribuir com os profissionais e com o sistema de saúde, como suporte para identificar o tipo de lesão e direcionar possibilidades de tratamento. Como benefícios indiretos pode-se obter informações precisas sobre indicadores para decidir sobre políticas públicas de saúde para tratamento de lesões, visando melhorar qualidade de vida do paciente e redução de custos para o sistema de saúde.

Baseado nesse problema e contexto, levantou-se a seguinte pergunta de pesquisa:

- I. É possível realizar o diagnóstico dos diferentes tipos de lesões de pele por meio de imagens dermoscópicas utilizando processamento de imagens e redes de DL para classificação com acurácia balanceada superior a 87 %?

1.1.1 Solução proposta

O trabalho em questão busca desenvolver um modelo por meio da implementação de métodos de processamento de imagens e algoritmos de DL para classificar diferentes tipos de lesões de pele delimitados pelo conjunto de dados selecionado.

As hipóteses levantadas para essa solução foram:

- I. É possível diagnosticar lesões de pele por intermédio de processamento de imagens e redes de DL para classificação com acurácia balanceada superior a 87%;
- II. É possível diferenciar os tipos de lesões de pele utilizando apenas imagens dermoscópicas.

1.1.2 Delimitação de Escopo

Esse projeto de pesquisa irá desenvolver um modelo de classificação automática de lesões de pele empregando processamento de imagens e DL utilizando a linguagem Python. Os tipos de lesões de pele que serão possíveis classificar, foram definidos pelo conjunto de dados escolhido.

De maneira que, o universo da pesquisa irá representar uma pequena amostra, delimitada pelo conjunto de dados selecionado, limitado aos disponibilizados no *The International Skin Imaging Collaboration* (ISIC). Uma vez que, não serão feitas aquisições de imagens por meio de instituições de saúde regional ou nacional.

Além disso, a validação do modelo construído irá ter sua validação por meio de um subconjunto de imagens do conjunto de dados escolhido. Dessa forma, serão comparados com modelos no estado da arte, treinados com o mesmo conjunto de dados, para averiguar a qualidade do modelo criado. Não serão feitos testes ou avaliações com médicos ou especialistas.

1.1.3 Justificativa

O diagnóstico e a detecção do câncer de pele são tradicionalmente realizados por meio de triagem manual e inspeção visual. Essas abordagens de inspeção visual e a triagem de imagens de lesões por dermatologistas é demorada, complexa, subjetiva e propenso a erros (YU et al. 2017; VESTERGAARD et al. 2008). Isso se deve principalmente a complexa natureza das imagens de lesões de pele pigmentadas (ESTEVA et al. 2017). Além de fatores, como pelos, vasos sanguíneos, presença de óleos, bolhas e outros ruídos nas imagens captadas que dificultam no diagnóstico (MAHONY et al., 2019).

Por meio de uma pesquisa bibliográfica sobre sistemas automáticos de diagnósticos no campo da saúde da pele, os estudos apontam para uma satisfatória acurácia, utilizando processamento de imagem e visão computacional na identificação de diversos tipos de doenças de pele, como por exemplo, no diagnóstico de melanoma (ROCHA et al., 2019) e sistemas de triagem de lesões de pele (NEGRENTE, 2020; SANTOS, 2019). Esses modelos têm auxiliado os profissionais de saúde a tomarem decisões mais rápidas e precisas no diagnóstico e detecção do câncer de pele (MOBINY et al. 2019; ADEGUN e VIRINI, 2021).

Atualmente, os algoritmos de DL têm se tornado a abordagem padrão para modelos automatizados de diagnóstico baseado em imagens dermatoscópicas (ADEGUN e VIRINI, 2021). De maneira, que empregam arquiteturas de redes neurais convolucionais profundas (do inglês, Deep Convolutional Neural Networks, DCNN) com o processo de TL e obtiveram excelentes resultados, chegando a ultrapassar a acurácia de especialistas (MAHAJAN et al., 2018).

Realizando um comparativo com técnicas clássicas de classificação, que incluem: Árvore de Decisão, Métodos de Classificação Fuzzy e Support Vector Machine (SVM) (MOHAN

e DHARAN, 2019). Em comparação com técnicas de classificação de DL (ROCHA et al., 2019; NEGRETE, 2020; SANTOS, 2019). O desempenho de algoritmos de DL é superior em comparação com as técnicas clássicas. Isso ocorre devido a capacidade de aprendizado e extração de características profundas e hierárquicas em conjuntos de dados de imagens (GARG e SINGH, 2020). Além disso, os algoritmos de DL podem passar por processos de afinamento, com o uso de TL, que tendem a elevar a acurácia final do modelo utilizando uma quantidade menor de processamento computacional. Levando a uma minimização dos custos do diagnóstico e alcançando melhores resultados (MAHONY et al., 2019).

Portanto, apesar dos estudos anteriores nesta temática, identifica-se ainda uma oportunidade de avaliar o desempenho em um contexto mais amplo e, acessível. Além disso, torna a operação mais eficientes, dessa maneira, fornecendo aos profissionais de saúde um apoio ao diagnóstico que permita rapidamente sugerir tratamentos e direcionar com agilidade casos graves para especialistas, potencialmente aumentando a chance de cura e recuperação do paciente. Dessa forma, reduzindo a taxa de mortalidade causadas pelo câncer de pele e suas consequências.

A justificativa se baseia em desenvolver um modelo que auxilie os profissionais da saúde, de maneira rápida; e com alta acurácia e revocação, capaz de aplicar métodos e técnicas de processamento de imagens e visão computacional a fim de realizar o diagnóstico dos tipos de lesões de pele de forma automática, dessa forma, agilizando e melhorando o atendimento ao paciente. Assim como, obtendo benefícios indiretos na redução de custos para o sistema de saúde e diminuindo sua sobrecarga de atendimentos.

1.2 OBJETIVOS

Esta seção formaliza os objetivos do trabalho, conforme descrito a seguir.

1.2.1 Objetivo Geral

Diagnosticar os tipos de lesões de pele presentes no conjunto de dados selecionado, analisando imagens dermoscópicas, utilizando métodos e técnicas de processamento de imagens e redes de DL usados para a tarefa de classificação.

1.2.2 Objetivos Específicos

- Obter imagens dermoscópicas de lesões de pele no ISIC;

- Estabelecer para análise um procedimento comum as imagens dermoscópicas para a aplicação dos diferentes métodos e técnicas de processamento de imagens digitais;
- Categorizar os diferentes tipos de lesões de pele por meio de redes de DL; e
- Validar os resultados obtidos, comparando-os com o estado da arte.

1.3 METODOLOGIA

A metodologia utilizada para a composição deste trabalho envolveu uma pesquisa bibliográfica, sendo realizada uma seleção de livros, periódicos e sites que continham informações e descrições sobre os assuntos abordados, como diferentes tipos de lesões de pele, visão computacional, processamento de imagens digitais no campo da saúde e, por fim, redes neurais convolucionais e suas métricas de avaliação, além da apresentação de trabalhos similares.

Nas próximas subseções são descritos: (1.3.1) metodologia de pesquisa, abordando as especificações da pesquisa; e posteriormente (1.3.2) etapas dos procedimentos metodológicos seguidas neste trabalho.

1.3.1 Metodologia de Pesquisa

A natureza da pesquisa, pode-se considerar que é aplicada, pois tem por objetivo aplicar na prática uma solução para o problema proposto, de maneira que, será respondida à pergunta de pesquisa de forma quantitativa. Os objetivos serão cumpridos por meio de uma pesquisa exploratória, investigando o uso de técnicas de processamento de imagem que envolvem desde a aquisição até técnicas de classificação das imagens.

1.3.2 Procedimentos Metodológicos

O trabalho adota as seguintes etapas metodológicas:

- I. **Fundamentação teórica:** Consiste no levantamento do material de estudo, assim como, a descrição dos métodos, algoritmos e técnicas estudadas para o desenvolvimento do trabalho em questão. Como também, como será selecionado cada modelo de acordo com a sua métrica de avaliação.
- II. **Trabalhos similares:** Levantamento de trabalhos similares envolvendo o processamento e a classificação de lesões de pele utilizando arquiteturas de redes

neurais convolucionais profundas e TL com ênfase no estudo de casos do melanoma e quais métricas de avaliação devem ter uma maior relevância na comparação do estado da arte.

- III. **Desenvolvimento:** Descrição dos detalhes acerca da realização dos experimentos para o desenvolvimento do trabalho, como também, será descrito os experimentos em si.
- IV. **Considerações finais:** Um apanhando sobre os acontecimentos que ocorreram durante a execução do trabalho, assim como, quais as possíveis contribuições do trabalho, e por fim, possíveis trabalhos a serem feitos para a continuação desse.

1.4 ESTRUTURA DO TRABALHO

Este documento está estruturado em quatro capítulos, sendo: (i) Introdução; (ii) Fundamentação Teórica; (iii) Desenvolvimento; e (iv) Considerações Finais.

No Capítulo 1, Introdução, é apresentada uma visão geral do trabalho, incluindo: problematização, objetivos do projeto, metodologia utilizada e a estrutura do trabalho propriamente dita.

Já o Capítulo 2, Fundamentação Teórica, é constituído por uma revisão bibliográfica sobre os conceitos que constituem a formação do projeto. Dessa forma, é feita uma descrição, sobre os conceitos básicos de imagens digitais e médicas; como também, sobre os dados utilizados no trabalho e sua apresentação; são abordadas sobre as técnicas de processamento de imagens que serão utilizados; assim como, é demonstrado os conceitos sobre rede neural, passando pela arquitetura básica de uma rede neural artificial e uma rede neural convolucional até as arquiteturas de redes neurais convolucionais profundas. Ao final, a sessão dos trabalhos similares.

No Capítulo 3, Desenvolvimento, são detalhados como e com o que serão realizados os experimentos. Assim como, como será cada experimento e como eles serão avaliado.

Por fim, o Capítulo 4, considerações finais, são abordados sobre realizações que ocorreram ao longo do desenvolvimento desse trabalho, dificuldades que ocorreram, problemas e suas soluções.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os conceitos e definições relevantes para a compreensão dos aspectos necessários para o desenvolvimento da solução proposta nesse trabalho. O capítulo está dividido em cinco seções principais, aprofundando-se nas definições de imagens digitais, sobre os dados utilizados, processamento de imagens, classificação, e por fim, trabalhos similares.

2.1 Imagem digital

Nessa seção, será apresentada conceitos sobre a imagem digital, assim como, será abordado sobre imagens médicas.

2.1.1 Representação

Segundo GONZALEZ e WOODS (2018), uma imagem pode ser definida como uma função bidimensional, $p(s, d)$ em que s e d são coordenadas espaciais (plano), e a amplitude de p em qualquer par de coordenadas é chamada de intensidade ou nível de cinza da imagem. Cabe observar que, uma imagem é composta de um número finito de elementos, de maneira que, cada elemento tem uma localização e um valor específico. Esses elementos recebem o nome de Pixels.

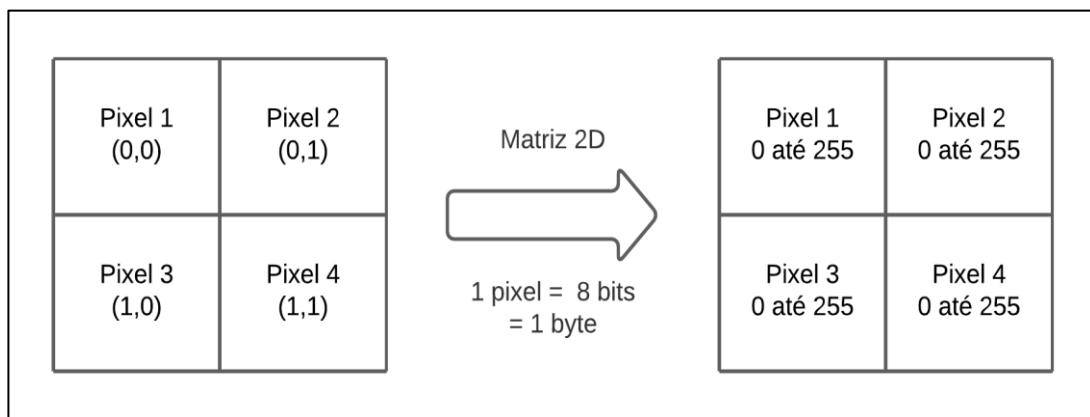
Dessa forma, uma imagem digital pode ser representada por uma matriz 2D, $p(s, d)$, contendo M linhas e N colunas. Onde, $s = 0, 1, 2, \dots, M-1$ e $d = 0, 1, 2, \dots, N-1$. Cabe ressaltar que, a seção do plano real que se expande pelas coordenadas de uma imagem é chamada de domínio espacial, com s e d sendo chamadas de variáveis espaciais e coordenadas espaciais. A notação matricial é expressa na Equação 1. Claramente, $a_{ij} = p(s = i, d = j) = p(i, j)$ (GONZALES; WOODZ, 2018).

$$A = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \cdots & \cdots & \cdots & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (1)$$

2.1.1.1 Escala cinza

As imagens na escala cinza apresentam duas dimensões, sendo cada elemento chamado de pixel. De maneira que, cada pixel pode receber um valor entre 0 e 255 (8 bits). Isso é, o valor 0 é denotado preto e o 255 é denotado branco. Entre esses dois extremos, estão os tons de cinza presentes nessa escala. Na Figura 1 é representada uma imagem 2x2px na escala cinza (GONZALEZ; WOODS, 2018).

Figura 1. Imagem 2x2px em escala cinza



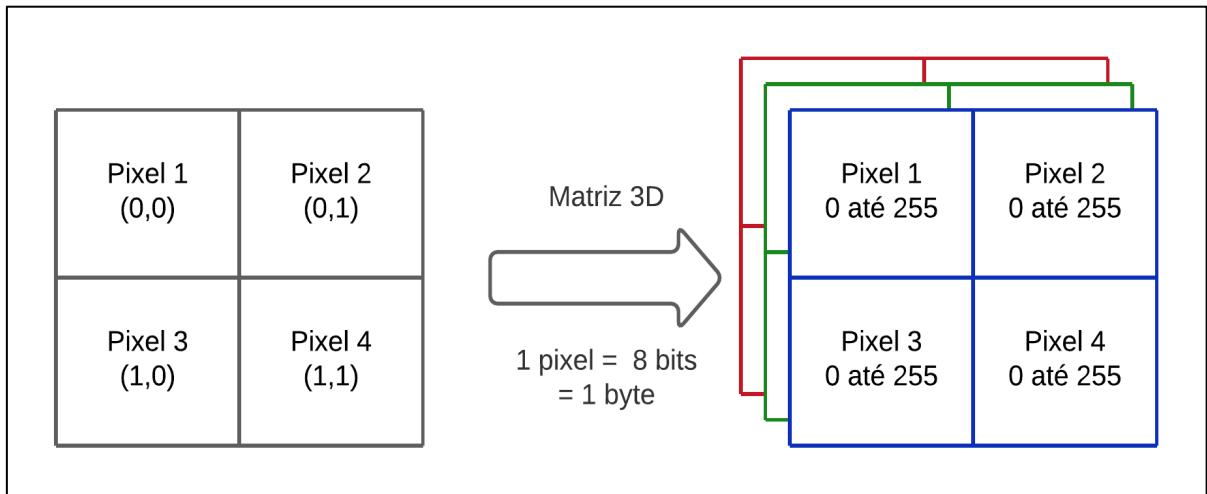
Fonte: O autor.

2.1.1.2 Escala colorida

As cores adicionam uma dimensão a mais em comparação com a escala cinza, tendo um total de três dimensões. Em virtude disso, cada uma das camadas da terceira dimensão representando a sua determinada cor. Por conseguinte, ao juntar a informação das três matrizes bidimensionais, obtém-se qualquer cor do espectro visível, isto é, uma imagem colorida composta por três imagens individuais, referidas às componentes primárias vermelho, verde e azul, segundo o modelo RGB ('Red – Green – Blue'). De modo que, cada pixel de cada cor funciona da mesma maneira que uma imagem em escala cinza (GONZALEZ; WOODS, 2018).

O modelo RGB apresentado na Figura 2, é somente um dos modelos de cores (espaço de cores ou sistema de cores) que existem. Essencialmente, um modelo de cor é uma especificação de um sistema de coordenadas e um subespaço dentro desse sistema no qual cada cor é representada por único ponto (GONZALEZ; WOODS, 2018).

Figura 2. Imagem 2x2px em escala RGB



Fonte: O autor.

2.1.2 Imagens médicas

O ISIC, que constitui uma parceria entre a acadêmica e a industria, com o objetivo de colaborar para a facilitação da aplicação de imagens digitais de lesões de pele para ajudar a reduzir a mortalidade por melanoma. Já que, quando reconhecido e tratado em seus estágios iniciais, o melanoma é facilmente curável (ISIC, 2021).

Outro objetivo do ISIC é a padronização das imagens dermatologicas , sendo que, a falta de padrões para imagens dermatológicas prejudica a qualidade e a utilidade da imagem de lesões cutâneas. Em virtude disso, o ISIC está desenvolvendo padrões para abordar as tecnologias, técnicas e terminologia usadas em imagens de pele com atenção especial às questões de privacidade e interoperabilidade (ou seja, a capacidade de compartilhar imagens entre plataformas de tecnologia e clínicas) (ISIC, 2021).

Com a possibilidade de classificar imagens, abriu-se uma nova possibilidade para a análise de imagens médicas. Atualmente os melhores modelos de classificação de imagens médicas são derivados da utilização de DCNN. Os resultados apresentados desafiam fortemente a acurácia e a rapidez de um especialista (LITJENS et al., 2017).

Além disso, o ISIC desenvolveu e está expandindo um arquivo de acesso público de código aberto de imagens de lesões de pele para testar e validar os padrões propostos. Este arquivo serve como um recurso público de imagens para o ensino e para o desenvolvimento e teste de sistemas de diagnóstico automatizados (ISIC, 2021).

Por fim, com o avanço das tecnologias mundiais, uma importante área que está em desenvolvimento e irá mudar completamente a maneira ao qual várias organizações devem funcionar, como também, deve influenciar na forma ao qual coletam seus dados é a chamada Internet das coisas, mais conhecida como *Internet of Things (IoT)* que poderia conectar diversos equipamentos médicos em diversos hospitais e criar uma coleção *big data*. Cada hospital poderia contribuir com seus dados é assim criar e aprimorar sistemas de diagnósticos automáticos usando imagens. Dessa forma, criando o que poderia ser uma subárea de IoT, a *The internet of Medical Things (IoMT)* (SERTE; SERENER; ALÂTURJMAN, 2020).

2.1.2.1 Imagens dermatoscópicas

Devido ao fato, que em diversos casos de diagnóstico de lesões, as lesões pigmentadas não são diagnosticadas por suas características clínicas, mesmo ao ser realizada por profissionais experientes, critérios adicionais são necessários para o diagnóstico clínico com uma maior acurácia. Dessa maneira, desenvolveu-se o método de exame auxiliar, a dermatoscopia (REZZE; SÁ; NEVES, 2006).

Esse método pode ter outros nomes, como: microscopia de superfície, dermoscopia, ou também, microscopia de epiluminescência. É um método utilizado para visualizar estruturas localizadas abaixo da camada mais externa da pele, indicado para o diagnóstico de lesões de pele pigmentadas. A técnica para a realização da dermatoscopia consiste no emprego de aparelhagem ótica que permite um aumento variável de seis a 400x (REZZE; PASCHOAL; HIRATA, 2014).

Por meio desse instrumento é possível aumentar a sensibilidade para detecção de lesões de pele pigmentadas evitando, assim, biopsias desnecessárias. A acurácia para o diagnóstico clínico do melanoma, feito por dermatologista, sem a utilização do dermatoscópio, foi estimada entre 75 e 80%, esse valor é mais baixo, se estabelecida por residentes de dermatologia ou clínicos gerais. Já com a utilização do exame dermatoscópico, pode-se alcançar acurácia diagnóstica de aproximadamente 90% para o diagnóstico de melanoma cutâneo (MENZIES *et al.*, 2001).

Com o passar do tempo e o avanço das tecnologias, a partir da dermatoscópio analógica, foi possível aprimorar os instrumentos que permitiram elevar a acurácia dos diagnósticos das lesões pigmentadas. Os aparelhos dermatoscópicos estão cada vez mais leves e de mais fácil manuseio, como o que está demonstrado na Figura 3. Aprimoraram-se também, os recursos para

realização da dermatoscopia digital, possibilitando o monitoramento de lesões pigmentadas ao longo do tempo, pelo armazenamento das imagens digitais (REZZE; PASCHOAL; HIRATA, 2014).

Figura 3. Dermatoscópio de polarização sem fio DE350, com ampliação óptica de 15x e 45x digital.



Fonte: Firefly (2021).

2.2 Dados

Nesta seção serão apresentados a origem dos dados utilizados, assim como, a explicação dos dados em si. Além disso, será discutido sobre cada categoria presente no conjunto de dados.

2.2.1 Dataset HAM10000

De acordo com Tschandl, Rosendahl e Kittler (2018), o conjunto de dados *Human Against Machine with 10000 training images* (HAM10000) é um dos maiores conjunto de dados para o treinamento de rede neurais para o diagnóstico automático de lesões de pele. Esse conjunto de dados foi popularizado por meio da utilização no desafio do ISIC 2018¹. As imagens foram coletadas com a aprovação do Comitê de Revisão de Ética da *University of Queensland* (Protocolo-No. 2017001223) e da *Medical University of Vienna* (Protocolo-No. 1804/2017).

As imagens dermatológicas foram adquiridas durante um período de 20 anos com uma variedade de tipos de dermatoscópios, em todas as áreas do corpo (exceto mucosa e unhas), de uma amostra histórica de pacientes apresentados para rastreamento de câncer de pele, de várias instituições diferentes (TSCHANDL; ROSENDAHHL; KITTLER, 2018).

¹ Link para acessar o desafio ISIC 2018: <https://challenge2018.isic-archive.com/>

O diagnóstico de cada imagem presente do HAM10000, foi realizado por meio de métodos como: Histopatologia; microscopia confocal de refletância; O acompanhamento por 2 anos das lesões; e por fim, a realização dos diagnósticos por três diferentes dermatologistas de cada imagem dermatoscópica presente no conjunto de dados. Foi possível a verificação e a construção de uma tabela verdade (do inglês, *ground truth*). De maneira que, essa tabela é utilizada internamente no treinamento, assim como, para a verificação da pontuação das métricas de avaliação (TSCHANDL; ROENDAHL; KITTLER, 2018).

Segundo Tschandl, Rosendahl e Kittler (2018), os autores do HAM10000, o conjunto de dados incluem exemplos representativos de lesões cutâneas pigmentadas que são praticamente relevantes. Mais de 95% de todas as lesões pigmentadas diagnosticadas durante a prática clínica será diagnosticada em uma das sete categorias de diagnóstico presentes no HAM10000.

Na prática, a tarefa clínica primária é diagnosticar entre uma lesão benigna e maligna, porém, também diagnosticar especificamente qual lesão se trata determinado caso. Pois, existem diferentes tipos de tratamentos para cada tipo de lesão. Principalmente as malignas, onde por exemplo, uma lesão melanoma e a carcinoma basocelular tem tratamentos totalmente diferentes. As lesões não pigmentadas, que são mais diversas e têm um maior número de diagnósticos diferenciais possíveis, não fazem parte deste conjunto (TSCHANDL; ROSENDAHL; KITTLER, 2018).

2.2.1.1 População amostral

As imagens dermatoscópicas do HAM10000, foram coletadas durante um período de 20 anos em dois locais diferentes, o Departamento de Dermatologia da Universidade Médica de Viena, Áustria, e a prática de câncer de pele de Cliff Rosendahl em Queensland, Austrália (TSCHANDL; ROSENDAHL; KITTLER, 2018).

O HAM10000 inclui lesões pigmentadas de diferentes populações, uma vez que, essa população, consiste em lesões de pacientes encaminhados a um centro de referência europeu terciário especializado na detecção precoce de melanoma em grupos de alto risco. Esse grupo de pacientes costuma apresentar um alto número de nevos e história pessoal ou familiar de melanoma. Como também, inclui lesões de pacientes de instalações de cuidados primários em uma área de alta incidência de câncer de pele.

De acordo com Tschandl, Rosendahl e Kittler (2018), os pacientes australianos são caracterizados por severos danos crônicos do sol. A pele cronicamente danificada pelo sol é caracterizada por vários lentigos solares e vasos ectáticos, que frequentemente estão presentes na periferia da lesão-alvo. Muito raramente, também pequenos angiomas e ceratoses seborreicas podem colidir com a lesão-alvo. De forma que, não foram removidos esses "ruídos" e os cabos terminais, pois, essa situação reflete a prática clínica.

Na maioria dos casos, embora nem sempre, a lesão-alvo está no centro da imagem. Imagens dermatoscópicas de ambos os locais de estudo foram obtidas por diferentes dispositivos usando dermatoscopia polarizada e não polarizada (TSCHANDL; ROENDAHL; KITTLER, 2018).

2.2.2 Categorias

O conjunto de dados tem como saída sete categorias sendo elas: Ceratoses actínicas e Carcinoma espinocelular (AKIEC); Carcinoma basocelular (BCC); Ceratose benigna (ceratose seborreica/ lentigo/ liquen plano) (BKL); Dermatofibroma (DF); Nevo melanocítico (NV); Melanoma (MEL); e por fim, Lesão Vascular (VASC) (TSCHANDL; ROENDAHL; KITTLER, 2018).

Por meio de uma primeira divisão, entre dois grandes grupos: não-melanocítico, que tendem a ter menos cores e arranjo arquitetural mais uniforme, maior homogeneidade e maior simetria; Já os melanocítico, tendem a ter o inverso das características citadas anteriormente (NUNO MENEZES, 2016).

Após essa primeira separação, uma segunda ocorreu. Essa segunda separação consistiu na divisão entre outros dois grupos. Foi definida a divisão entre as lesões que são benignas, que são tumores que tem seu crescimento de forma organizada, geralmente lento, expansivo e apresentam limites bem nítidos e apesar de não invadirem os tecidos vizinhos, podem comprimir os órgãos e tecidos adjacentes; e os tumores malignos que manifestam maior autonomia e são capazes de invadir tecidos vizinhos e provocar metástases, podendo ser resistentes ao tratamento e causar a morte do hospedeiro (INCA, 2011). Por fim, o resultado dessas separações é demonstrado no Quadro 1, seguido do nome da categoria dentro do HAM, suas principais características, e por fim, uma imagem da lesão.

Quadro 1. Comparativo entre as sete categorias presentes no HAM10000

Melanocítico		Não-Melanocítico				
Maligno	Benigna	Maligno		Benigna		
Melanoma	Nevo-Melanocítico	Carcinoma Basocelular	Ceratoses Actinicas e Carcinoma Espinocelular	Dermatofibroma	Leão Vascular	Ceratose Benigna
MEL	NV	BCC	AKIEC	DF	VASC	BKL
A lesão é assimétrica, bordas irregulares, entalhes ou borraduras, cor variável ou com distribuição irregular e diâmetro acima de 6cm.	A lesão é máculas, pápulas ou com nódulos pequenos, circunscritos; variam na cor desde azul/negro até marrom, cor-de-rosa ou cor de pele.	A lesão apresenta pápula brilhante com telangiectasia, ou também, placas eritematosas plana. A semelhanças com cicatrizes com limites mal definidos.	A lesão apresenta áreas avermelhadas ou ligeiramente acastanhadas, com uma superfície áspera e queratósica.	A lesão apresenta uma área central mais clara e um colarete de pigmentação mais escuro.	A lesão é caracterizada por uma coloração avermelhada ou roxa e estruturas sólidas e bem circunscritas.	O conjunto de lesões apresenta forma oval com bordas lisas recortadas com cor castanho-escuros ou claras

Fonte: Adapatado de Tschandl; Rosendahl E Kittler (2018).

2.3 PRÉ-PROCESSAMENTO

Nessa seção serão apresentados algoritmos, métodos e técnicas de pré-processamento que irão ajudar o modelo a classificar as imagens dermatoscópicas.

2.3.1 Constância de Cor

Ao se utilizar imagens que veem de múltiplas fontes em diferentes configurações de iluminação, essa falta de padrão entre as imagens, acaba causando uma redução no desempenho dos sistemas de classificação. Assim como, dificulta uma comparação justa com outros modelos. Já que eles podem utilizar diferentes conjunto de dados, vindo de diferentes instituições médicas com equipamentos e condições de iluminação diferentes (BARATA; CELEBI; MARQUES, 2014).

Segundo Barata, Celebi e Marques (2014), ao se utilizar técnicas de compensação de cor para reduzir a influência das configurações de iluminação presentes na imagem, irá fornecer uma melhora no desempenho da classificação. Como também, minimizar as diferenças nas

aquisições, favorecendo tanto o modelo a ser treinando, quanto ao se comparar com outros modelos (BARATA; CELEBI; MARQUES, 2014).

O objetivo dos algoritmos de constância de cor é transformar as cores de uma imagem I , adquirida sem a influência do iluminante, para que elas apareçam idênticas as cores sob uma fonte de luz canônica. Usualmente, é assumido que essa fonte de luz canônica é um iluminante branco ideal (BARATA; CELEBI; MARQUES, 2014).

As transformações de cores são realizadas em dois passos. Primeiro, o iluminante desconhecido é estimado no espaço de cor, por exemplo, para o sistema RGB. Dessa forma assume-se que o iluminante desconhecido no espaço de cor do sistema RGB é $[e_R \ e_G \ e_B]^T$. Como o segundo passo, a imagem é transformada usando as condições estimadas do iluminante desconhecido. Dessa maneira, diferentes algoritmos podem ser utilizados para estimar a condição do iluminante desconhecido (BARATA; CELEBI; MARQUES, 2014).

Dessa forma, a Equação 2 demonstra o princípio básico de aplicação do algoritmo, onde I_i é denotado como o i -ésimo componente da imagem I , $\mathbf{q} = (q, v)$ é a posição de cada pixel, e \mathbf{k} é uma constante de normalização que garante que $\mathbf{e} = [e_R \ e_G \ e_B]^T$ tenha um comprimento unitário em relação à norma euclidiana:

$$\left(\frac{\int (I_i(\mathbf{q}))^n d\mathbf{q}}{\int d\mathbf{q}} \right)^{1/n} = k e_i \quad (2)$$

O algoritmo *Shades of Gray* (FINLAYSON; TREZ, 2004) utiliza a distância de Minkowski Normalizada (Equação 3) para estimar o iluminante desconhecido. O parâmetro n é o grau dessa norma (BARATA; CELEBI; MARQUES, 2014):

$$D_p(q, v) = \left(\sum_{i=1}^n |q_i - q_i|^n \right)^n \quad (3)$$

Dessa forma, após a resolução da Equação 2, concluísse o primeiro passo do algoritmo, obtendo-se o estimador \mathbf{e} , o iluminante desconhecido. Para a transformação da imagem I , deverá se utilizar o modelo diagonal von Kries (KRIES, 1970):

$$\begin{pmatrix} I_R^t \\ I_G^t \\ I_B^t \end{pmatrix} = \begin{pmatrix} c_R & 0 & 0 \\ 0 & c_G & 0 \\ 0 & 0 & c_B \end{pmatrix} \begin{pmatrix} I_R \\ I_G \\ I_B \end{pmatrix} \quad (4)$$

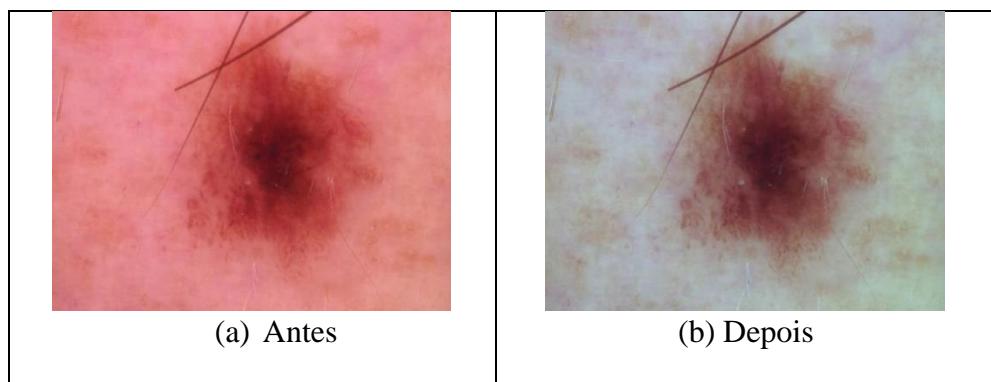
De maneira que, $[I_R, I_G, I_B]^T$ representa os valores dos pixels que foram adquiridos sem a influência do iluminante. Já $[I_R^t, I_G^t, I_B^t]^T$ representa os valores de pixels transformados, como eles devem aparecer sob o iluminante desconhecido. O que é assumido como o iluminante branco ideal, ou seja, $e^c = \left(\frac{1}{\sqrt{3}} \quad \frac{1}{\sqrt{3}} \quad \frac{1}{\sqrt{3}} \right)^T$ (BARATA; CELEBI; MARQUES, 2014).

Os coeficientes da matriz $\{c_R, c_G, c_B\}$ estão relacionados ao estimador do iluminante desconhecido (**e**), os coeficientes associados ao iluminante que geram a transformação (BARATA; CELEBI; MARQUES, 2014):

$$c_i = \frac{1}{\sqrt{3}e_i}, i \in \{R, G, B\} \quad (5)$$

A distância de Minkowski é o resultado normalizado do vetor estimado do iluminante ideal (FINLAYSON; TREZ, 2004). Pela mesma razão, ele calcula uma média ponderada dos valores de pixels, atribuindo pesos mais altos para pixels com intensidades mais altas. A função peso proposta é baseada na distância **n** do Minkowski. Além disso, através de experimentos repetidos, quando $1 < n < \infty$, Finlayson e Trezzi (2004), obtiveram um valor aproximadamente preciso, ou seja, quando $n = 6$, o algoritmo *Shades of Gray* realiza os melhores resultados de estimação do iluminante desconhecido. Por fim, na Figura 4 é demonstrada uma aplicação em uma das imagens presentes no HAM10000.

Figura 4. Exemplos de uma imagem antes e depois da aplicação do algoritmo *Shades of Gray* com a distância de Minkowski Normalizada igual a 6



Fonte: Tschandl; Rosendahl E Kittler (2018).

2.3.2 Aumento de dados

O aumento de dados (do inglês, *Data augmentation, DA*) é uma técnica capaz de atuar como um regularizador, como também, ajudar a reduzir o *overfitting* ao treinar um modelo de DL, e por fim, acaba melhorando a capacidade de generalização dos modelos. Essa técnica, consiste em aumentar a quantidade de dados adicionando cópias ligeiramente modificadas de dados já existentes ou adicionando dados sintético recém-criados a partir de dados existentes. (CONNOR; TAGHI, 2019).

As técnicas que realizam as modificações nas imagens, consistem em aplicar, uma probabilidade aleatória e/ou algumas técnicas utilizam um fator, normalmente um intervalo de valores. Essas técnicas são: brilho, contraste, saturação, nitidez, ruído ou brilho aleatórios nas imagens. Como também, aplicar giros, cortes, distorções, tesouramento e entre outros. Dessa forma, criando uma imagem nova para o conjunto de dados, como demonstrado na Figura 5.

Figura 5. Exemplos de aplicações de técnicas de aumento de dados em uma imagem DF



Fonte: Sae-Lim; Wettayaprasit e Aiyarak (2019).

Evidentemente, pode-se verificar que DA é uma das chaves essenciais para superar o desafio de dados de treinamento limitados por "aumentar" aleatoriamente a diversidade e número de dados. Assim, a construção das *pipelines* (técnicas sucessivas de aumento de dados) requer experiência para cada tipo de imagem em diferentes campos.

2.3.3 Redimensionamento e Reescalonamento

Os modelos pré-treinados são normalmente treinados com imagens com um tamanho fixo, ao passo que esse tamanho é significativamente menor do que quando as imagens de lesões de pele são capturadas e, consequentemente, as imagens dermatoscópicas são reduzidas. Todavia, informações médicas úteis podem ser perdidas durante esta transformação. Esse fato ocorre, devido às limitações computacionais, que tornam impossível ajustar as redes com uma imagem de alta resolução, como as que as imagens dermatoscópicas apresentam (MAHBOD et al, 2020).

Nesse caso, um redimensionamento ideal é algo necessário para minimizar as perdas de informações e alcançar bons desempenhos nas redes de classificação. Um estudo realizado por Mahbod et al. (2020), demonstrou que, ao se utilizar imagens muito pequenas, por exemplo, de tamanho 64×64 pixels ou menor, o desempenho se degrada, afetando o processo de classificação, enquanto, para imagens de tamanho médio igual a 128×128 pixels ou maior, ocasionaram em uma rede com um desempenho relativamente superior as das imagens pequenas. De maneira que, as imagens com tamanhos superiores as imagens médias, apresentaram uma classificação ligeiramente melhor (MAHBOD et al, 2020).

Já para caso do reescalonamento dos valores das imagens para o treinamento dos modelos pré-treinados, esperasse que os dados de entrada estejam em um tipo específico de padronização, sendo que inicialmente os dados estão em uma faixa entre 0 e 255 em cada pixel, como explicado na primeira seção desse capítulo. Como esses valores não são ideais para os modelos, eles passam por uma função de padronização dos valores, a camada de reescalonamento.

Por exemplo, sua escala muda para uma faixa entre 0 e 1 ou -1 e 1. Ou seja, os dados ficam com a mesma escala e variância. Ademais, o reescalonamento reduz a complexidade da rede. Assim como, aumenta potencialmente a acurácia e ajuda na velocidade de treino do modelo. Por fim, cada modelo pré-treinado apresenta sua forma de padronização como descritas na API do Keras. (CHOLLET, 2018).

2.4 CLASSIFICAÇÃO

Nessa seção, inicialmente serão apresentados os conceitos sobre DL, seguido por conceitos básicos sobre neurônios artificiais, rede neural artificial, rede neural convolucional, TL,

as arquiteturas de rede neural convolucional profundas pré-treinadas, conceitos de funções e otimizadores, e por fim, algoritmos utilizados no treinamento.

2.4.1 Aprendizado Profundo

Com o desenvolvimento do diagnóstico auxiliar por computador por meio de inteligência artificial está sendo possível fornecer uma maneira promissora de tornar o processo de diagnóstico mais eficiente e disponível para as massas. Nesse quesito, o aprendizado profundo (do inglês, *Deep Learning*, DL) é a subárea de inteligência artificial que lida com uma ampla gama de tarefas, incluindo problema de imagens médicas (VUPPALA et al., 2020).

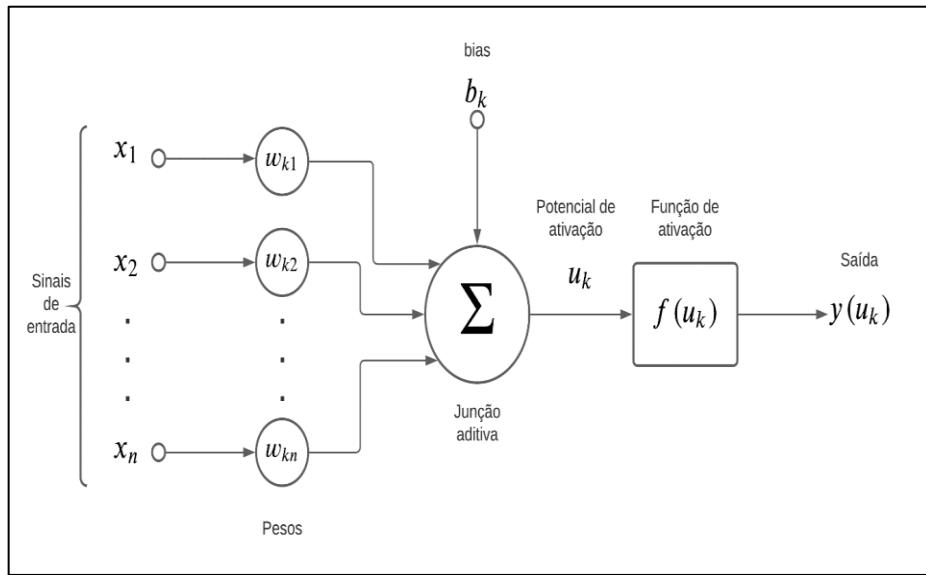
DL, é uma subárea de aprendizado de máquina (do inglês, *Machine Learning*, ML), assim como, ML é de inteligência artificial, que exploram muitas camadas de processamento de informações não lineares para extração e transformação de recursos supervisionados ou não supervisionados e para análise e classificação de padrões (DENG; YU, 2014). É um conjunto de algoritmos em aprendizado de máquina que tenta aprender recursos importantes de dados brutos automaticamente. Como as redes neurais tradicionais, as redes neurais profundas também consistem em neurônios artificiais organizados na forma de camadas de entrada, ocultas e de saída (AHMAD; FARMAN; JAN, 2019).

2.4.2 Neurônio artificial

Um neurônio artificial (**Erro! Fonte de referência não encontrada.**) nada mais é que uma unidade de processamento de informação que é fundamental para a operação de uma rede neural (que será apresentada na seção seguinte) (HAYKIN, 2008).

De maneira que, os sinais de entradas são representados por um conjunto de atributos $x = \{x_1, x_2, \dots, x_n\}$, sendo n , o número total de entradas. Já o conjunto $w = \{w_{k1}, w_{k2}, \dots, w_{kn}\}$, onde k é o número dos neurônios, são chamados de pesos sinápticos, cada conexão entre dois neurônios tem um peso associado, que é um valor numérico que representa a força da conexão entre os dois neurônios. Ao realizar uma soma ponderada entre os sinais de entrada e seus respectivos pesos, ou seja, $\sum_{i=0}^n x_i \times w_{ki}$, irá resultar em um potencial de ativação, denominado u_k (HAYKIN, 2008).

Figura 6. Anatomia do neurônio artificial



Fonte: Adaptado de Haykin (2008).

De modo que, no somador à adição de um *bias*, que será aplicado externamente, representado por b_k . O *bias* tem como função, aumentar ou diminuir o valor da função de ativação, dependendo do sinal dele, tal como, aumentar o grau de liberdade da função. Por fim, com o potencial de ativação junto ao *bias*, a passagem dessa função por uma função de ativação (f), que resulta em uma saída representada por y no neurônio artificial. De maneira que, $y_k = f(u_k)$, em termos matemáticos um neurônio artificial pode ser representado pela Equação 6 (HAYKIN, 2008):

$$y_k = f(\sum_{i=0}^n x_i \times w_{ki} + b_k) \quad (6)$$

Resumidamente, para cada entrada do neurônio, a um peso associado, de modo que, esse peso tem a função de aumentar a inclinação da função de ativação, modifica a rapidez com que a função de ativação será ativada. Enquanto o *bias*, há somente um por neurônio, e tem uma relação de intercepto, no mesmo sentido de uma equação linear. Ele irá ajustar a saída, de maneira, a ajudar o modelo a se adaptar melhor aos dados, ou seja, a sair da origem e se movimentar pelo eixo em função dos dados fornecidos. Como também, terá uma influência no controle do valor no qual a função de ativação será ativada. Dessa forma, se a saída ou ativação, excede certo *threshold*, controlado pelo *bias*, ocorre um ‘disparo’ do neurônio.

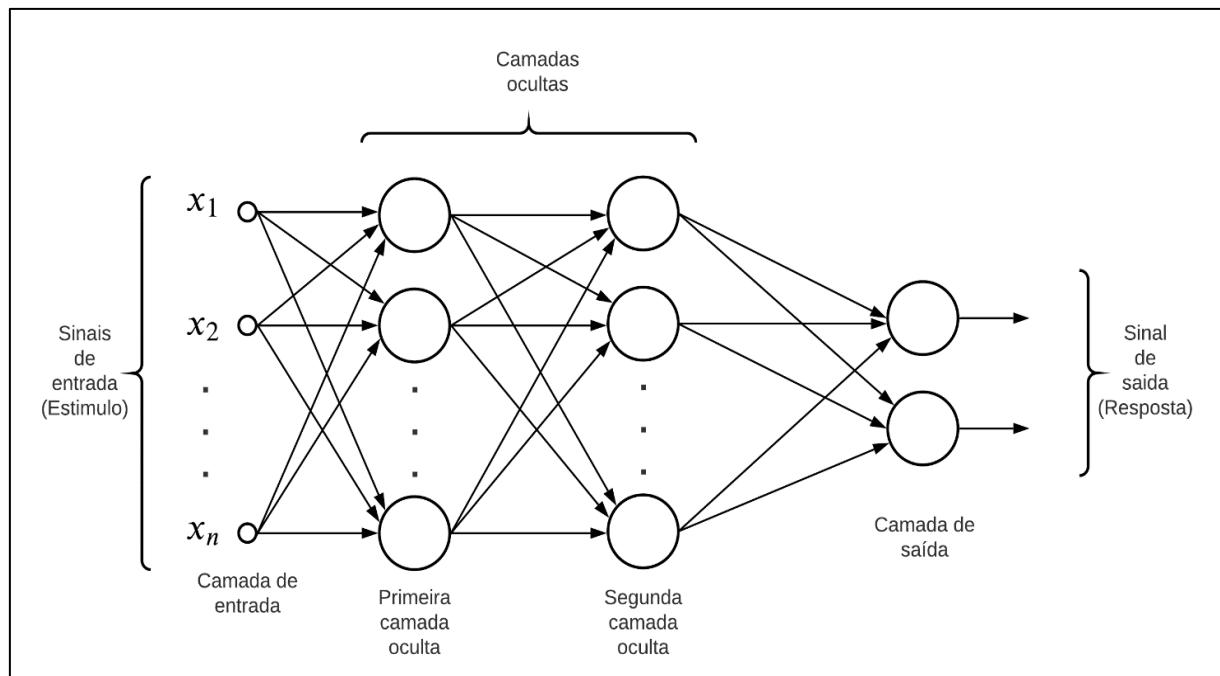
O processo de aprendizado é chamado de algoritmo de aprendizagem, cuja sua principal função é alternar diversas vezes os pesos sinápticos de cada neurônio presente em uma rede de uma forma ordenada para alcançar um objetivo de projeto desejado (HAYKIN, 2008).

2.4.3 Rede neural artificial

Segundo Haykin (2008), uma rede neural artificial (RNA) é formada por uma camada de entrada, uma ou mais camadas ocultas (caso tenha três ou mais camadas ocultas, a rede passa a ser chamada de rede profunda) e uma camada de saída. O sinal de entrada irá se propagar para frente por meio da rede, de onde se deriva seu nome *feedforward*, camada por camada.

Como demonstrado na Figura 7, a rede está totalmente conectada, um neurônio, em qualquer camada, está totalmente conectado com os neurônios (nós) da camada anterior. Dessa forma, para deixar explícito, o fluxo do sinal se propaga da esquerda para a direita e de camada há camada até a saída. Esse sinal é denominado sinal funcional, pois, ele deve realizar uma função útil na saída de rede, como também, a cada neurônio que esse sinal passa, ele é calculado como uma função de suas entradas e pesos associados no determinado neurônio (HAYKIN, 2008).

Figura 7. Grafo arquitetural de um perceptron de múltiplas camadas com duas camadas ocultas



Fonte: Adaptado de Haykin (2008).

A RNA tem diversas aplicações com sucesso, de maneira que, conseguem resolver diversos problemas difíceis. Por meio, de suas camadas ocultas, que não são partes nem da entrada e nem da saída, eles têm a capacidade de aprender tarefas complexas extraiendo progressivamente as características mais significativas dos sinais de entrada (HAYKIN, 2008).

2.4.4 Rede neural convolucional

As redes neurais convolucionais (CNN, do inglês, *Convolution Neural Network*) são um tipo particular de RNA com inspiração neurobiológica na organização do córtex visual dos animais apresentada nos trabalhos de Hubel e Wiesel (1962) e Hubel e Wiesel (1977). De forma similar a organização do córtex visual, as CNNs são projetadas para reconhecer padrões, sendo favorecidas por dados organizados em grade (RAMPAZZO, 2019).

Apresentada por LeCun e colaboradores em 1989, a camada convolucional recebe este nome por se basear na aplicação da função matemática denominada convolução (LECUN et al., 1998). A operação de convolução, um tipo específico de operação linear, é responsável por dar nome às CNNs, que podem ser consideradas RNAs nas quais a operação multiplicação geral de matrizes é substituída pela operação de convolução em pelo menos uma de suas camadas (GOODFELLOW; BENGIO; COURVILLE, 2016).

2.4.4.1 Camada de Convolução

A camada de convolução tem como função, aprender padrões locais em imagens ou em qualquer dado organizado em grade, como por exemplo, imagens e áudios. O uso de camadas convolucionais em imagens tem duas características interessantes, as quais são:

- Os padrões que são aprendidos são invariantes para as translações. Ou seja, quando um padrão é reconhecido em uma determinada imagem, em um certo local da imagem, esse padrão pode ser reconhecido em diferentes locais em outras imagens. Porém, as camadas convolucionais necessitam de treinamento com uma alta amostragem de dados para ter uma alta generalização (CHOLLET, 2018).
- Os padrões hierárquicos espaciais são aprendidos pelas camadas convolucionais, de maneira que, cada camada pode aprender diferentes padrões. Por exemplo, em uma primeira camada convolucional ela pode aprender sobre os pequenos padrões nas bordas e em uma segunda camada convolucional, pode aprender maiores padrões fornecidos pelas características aprendidas pela camada anterior, e assim por diante (CHOLLET, 2018).

A operação de convolução é formada por uma entrada e um filtro (do inglês, *kernel*). A saída dessa operação de convolução é referida como mapa de características. Em sua forma

geral, a operação de convolução acontece entre duas funções (f e g), de maneira que, gera uma terceira função expressa como a forma de uma modificação pela outra. A operação entre f e g é escrita como $f * g$. Onde, f é a entrada e g o filtro (RAMPAZZO, 2019).

Dada uma imagem como dado de entrada, a convolução opera com tensores 3D, com dois eixos espaciais (altura e largura), como também, um eixo de profundidade (também chamado de eixo de canais). Para uma imagem RGB, o eixo de canais recebe três valores, referente aos três canais de cores: vermelho, verde e azul. Já para uma imagem preto e branco, esse valor será um. A operação de convolução extraí os fragmentos da entrada e aplica transformações para todos esses fragmentos, produzindo assim, um mapa de características (CHOLLET, 2018).

O mapa de características continua sendo um tensor 3D, com a sua altura e largura. Porém, a sua profundidade é arbitrária, pois, a profundidade passa a ser um parâmetro da camada de convolução e não mais uma característica do número de canais da imagem de entrada. Agora ela é o número de filtros. Os filtros codificam aspectos específicos dos dados de entrada. Por exemplo, em um único filtro é possível codificar o conceito da “presença de um rosto na imagem” (CHOLLET, 2018).

Como exemplo, para uma imagem I bidimensional de entrada e um filtro K também bidimensional, a operação de convolução entre eles é representada pela Equação 7 (GOODFELLOW; BENGIO; COURVILLE, 2016):

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (7)$$

Porém, em bibliotecas computacionais para facilitar a implementação, geralmente é empregado a operação de correlação cruzada, dada pela Equação 8, como sinônimo da convolução. Embora a correlação cruzada não apresente a propriedade comutativa, como a operação de convolução, ambas são equivalentes para sinais simétricos (VON ZUBEN; VERONEZE, 2019). Normalmente, a correlação cruzada é aplicada na área de aprendizado profundo (RAMPAZZO, 2019).

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \quad (8)$$

A convolução discreta pode ser vista como multiplicação por uma matriz (GOODFELLOW; BENGIO; COURVILLE, 2016), dessa forma, a Figura 8, representa graficamente uma operação de correlação cruzada para uma entrada $I_{i,j}$, contendo 4 linhas de largura e 4 de altura;

e um filtro $K_{i,j}$, contendo 2 linhas de largura e 2 de altura, é deslizado através da matriz de entrada. O mapa de características é o resultado da operação. Neste exemplo, o passo (do inglês, *stride*) é de 2×2 , isso faz com que a janela do kernel movimente-se dois passos por vez.

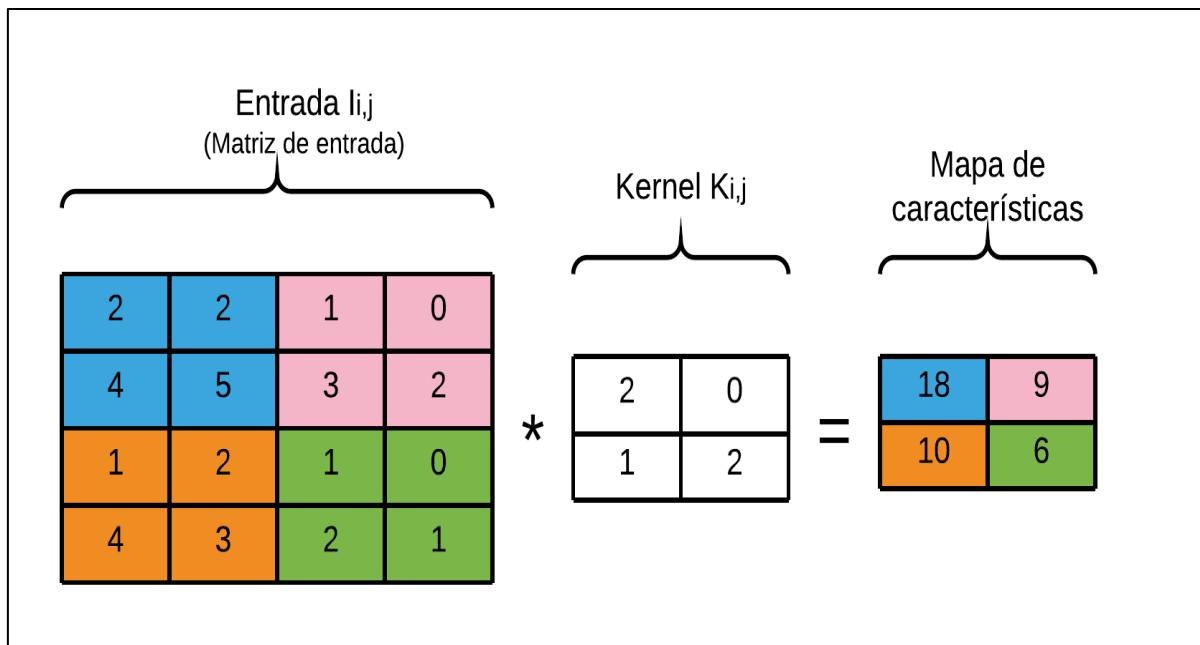
Dado como exemplo o primeiro elemento do mapa de característica, a Equação 9 dá a resposta:

$$S(2, 2) = \sum_{i=1}^n \sum_{j=1}^m X_{i,j} K_{i,j} \quad (9)$$

$$S_{1,1} = X_{1,1}K_{1,1} + X_{1,2}K_{1,2} + X_{2,1}K_{2,1} + X_{2,2}K_{2,2}$$

$$S_{1,1} = (2 \times 2) + (2 \times 0) + (4 \times 1) + (5 \times 2) = 18$$

Figura 8. Representação gráfica da operação de correlação cruzada para uma entrada 2D.



Fonte: O autor.

2.4.4.2 Camada de Pooling

Segundo Goodfellow, Bengio e Courville (2016), uma camada típica de uma rede convolucional consiste em três estágios. No primeiro, uma camada de convolução; no segundo, uma função de ativação não linear, como por exemplo a ReLU; e por fim, a camada *pooling* que irá modificar ainda a mais a saída.

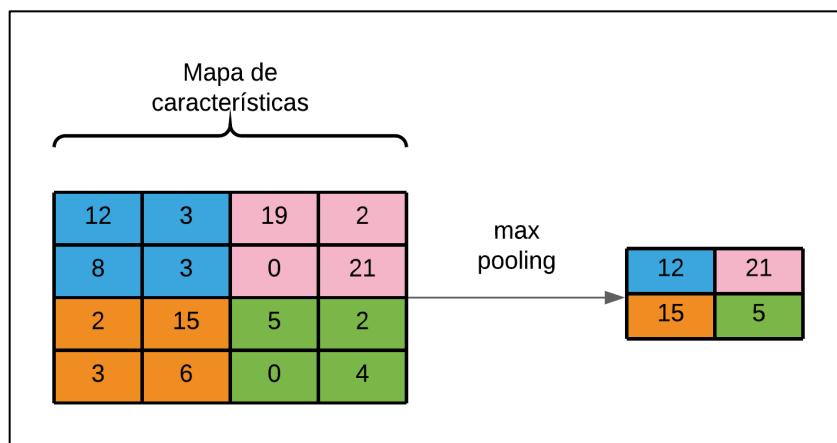
O *pooling* é importante para tornar os dados invariantes a pequenas translações da entrada, isto é importante para que a rede neural consiga reconhecer padrões das imagens em

diferentes posições (DUMOULIN; VISIN, 2016). É uma técnica utilizada para reduzir a dimensionalidade das imagens obtida da camada convolucional anterior, reduzindo o número de *pixels* na saída. Além, de reduz a quantidade de parâmetros no futuro e, portanto, a carga computacional.

A Figura 9, demostrada um exemplo aplicado de *max pooling*, essa aplicação emprega uma janela deslizante de dimensão 2×2 e *stride* 2×2 , o qual escolhe entre os valores presentes na janela o valor máximo do mapa de característica, avaliando entre todos os vizinhos.

O *max pooling* consiste em extrair por meio de uma janela, os valores máximos presentes no mapa de característica em cada canal. É conceitualmente semelhante à convolução, exceto que, em vez de transformar partes locais por meio de uma transformação linear aprendida (o *kernel* de convolução), eles são transformados por meio de uma operação de tensor máxima codificada (CHOLLET, 2018).

Figura 9. Calculando os valores de max pooling



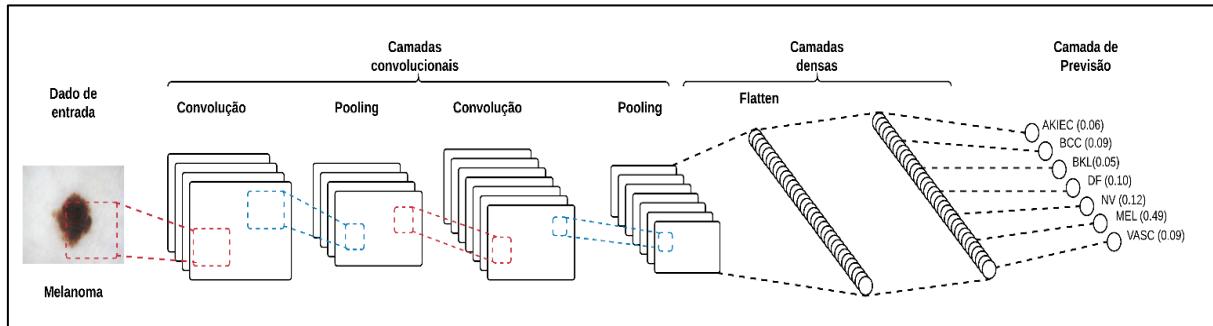
Fonte: O autor.

Além do *max pooling* que captura o valor máximo dentro da janela específica, existem o *average pooling* que captura o valor médio entre os elementos presentes dentro da janela, e por fim, o *global pooling*, que reduz cada canal do mapa de característica a um único valor. Ou seja, todo o mapa de característica, se transforma em um único valor. O *global pooling*, pode ser tanto do tipo *max*, quanto do tipo *average*.

2.4.4.3 Arquitetura típica da CNN

Com base nas seções anteriores, foi possível ver individualmente cada parte de uma rede neural convolucional, assim, foi possível analisar e compreender cada etapa que torna possível uma CNN, por exemplo, classificar imagens. Assim, juntando todos os conceitos das seções anteriores é possível obter o exemplo de uma arquitetura típica de CNN na Figura 10.

Figura 10. Exemplo de uma arquitetura típica CNN



Fonte: O autor.

A Figura 10, mostra uma rede neural convolucional que recebe uma imagem RGB, ou seja, uma imagem tridimensional, contendo largura, altura e o número de canais (no caso do RGB, esse valor seria 3). Essa imagem passa pelos filtros convolucionais, funções de ativação (como por exemplo, a ReLU) e as camadas de *pooling*. Concluindo assim, o processo de criação do mapa de característica.

Após a última camada de convolução ou *pooling*, há um processo de achatamento, conhecido como camada *flatten*, ou seja, um nivelamento de um tensor 3D para 1D. Dessa forma, a dimensão dessa camada é dada pela seguinte forma:

$$\text{Altura atual da imagem} \times \text{Largura atual da imagem} \times \text{Número de filtros.}$$

Seguido de uma camada densa ou também conhecida como totalmente conectada, que segue os mesmos princípios de uma RNA para o processo de classificação. Por fim, uma camada de previsão, que tem as dimensões definidas pelo número de categorias a serem classificadas. Usando o HAM10000 como exemplo, essa camada teria um total de sete neurônios. Como se trata de um problema de classificação multicategórica, a função de ativação presente nessa camada seria a Softmax.

2.4.5 Funções e otimizadores

Nessa subseção será abordado os conceitos de funções de ativação e custo, assim como, de qual maneira o otimizador usa a função de custo para alcançar a melhor direção para obter o menor erro e a melhor generalização do modelo na etapa de treinamento.

2.4.5.1 Função de ativação

De acordo com Haykin (2008), a função de ativação representada por $f_a(u_i)$, define a saída de um neurônio em termos do campo local induzido u_i (HAYKIN, 2008). As funções de ativação podem ser tanto lineares, quanto não lineares. Porém, em geral essas funções trazem componentes não lineares em sua saída. De maneira que, a presença de características dinâmicas nas propriedades envolvidas na definição do estado de ativação tem como consequência a aplicação de equações diferenciais. Caso contrário, não poderiam funcionar durante a retropropagação (do inglês, Backpropagation) dos sinais de ativação, ou sejam as funções de ativação são usadas no fim de um neurônio para introduzir complexidades não lineares ao modelo (SILVA; ZUBEN, 1998).

As funções de ativação (FAs) são responsáveis por realizar o processamento dos parâmetros livres calculando a entrada da camada subsequente e saída da rede neural. Além disso, as FAs remediam os problemas do desaparecimento e explosão do gradiente (do inglês, vanishing and exploding gradients) (NWANKPA et al., 2018).

Tal qual, se os resultados dos termos derivados que são geralmente menores que um, consequentemente, com a multiplicação sucessiva destes resultados, o valor torna-se cada vez menor e tende à zero, reduzindo a ação do gradiente e, portanto, o gradiente desaparece. Por outro lado, se os valores forem maiores que um, a multiplicação sucessiva aumentará os valores e o gradiente tende ao infinito, ultrapassando os limites do gradiente e, assim, explodindo o gradiente. Dessa forma, as FAs mantêm os valores desses gradientes para limites específicos (NWANKPA et al., 2018).

As funções de ativação descritas a seguir, são algumas das mais utilizadas para as redes neurais profundas que envolvam classificação de imagens categórica:

- **Rectified Linear Unit (ReLU)**

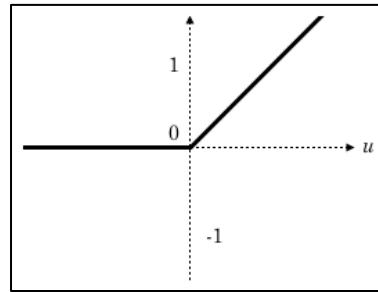
O ReLU, é uma das funções de ativação mais amplamente utilizada, pois tem um baixo custo computacional, de maneira que, apesar da simplicidade das operações realizadas obtém bons resultados (NWANKPA et al., 2018). Como também, possui uma grande capacidade de generalização da aprendizagem profunda (DAHL; SAINATH; HINTON, 2013) (ZEILER et al., 2013).

O ReLU representa uma função linear e, portanto, preserva as propriedades dos modelos lineares que os tornam fáceis de otimizar com o método do Gradiente Descendente, executa uma operação de limite para cada elemento de entrada em que valores menores que zero são definidos como 0, a função ReLU é expressa pela Equação 10 (NWANKPA et al., 2018):

$$f_a(u_k) = \max(0, u) = \begin{cases} u_i, & \text{se } u_k > 0 \\ 0, & \text{se } u_k \leq 0 \end{cases} \quad (10)$$

A função ReLU é geralmente aplicada nas camadas ocultadas das redes profundas com outra função de ativação aplicada nas camadas de saída da rede, como por exemplo, a função *Softmax*. O gráfico da função de ativação ReLU, pode ser demonstrado na Figura 11:

Figura 11. Função de ativação ReLU



Fonte: PAULY, Leo et al. (2017).

- ***Softmax***

A função de ativação *Softmax* é normalmente utilizada em modelos categóricos, pois, ela retorna uma probabilidade de cada classe, com a classe verdadeira para determinado mapeamento, tendo a maior probabilidade. Geralmente, essa função de ativação aparece na camada de saída das redes (NWANKPA et al., 2018).

Como mencionado acima, ela calcula a probabilidade de distribuição de um vetor de números reais. A função, produz uma saída que tem os limites entre 0 e 1, com a soma das

probabilidades sendo igual a 1. Dessa forma, a função *Softmax* pode ser calculada da seguinte forma, pela Equação 11 (NWANKPA et al., 2018):

$$f_a(u_k) = \frac{e^k}{\sum_j e^{u_j}} \quad (11)$$

2.4.5.2 Função de custo

A função de custo tem como objetivo, medir a performance entre os dados de entrada e os resultados das saídas da rede neural. Dessa forma, a função de custo controla os valores de saída, medindo o quanto longe a rede está do valor esperado (CHOLLET, 2018). Ou seja, a função de custo é a chave para ajustar os pesos de uma rede neural.

De maneira que, durante o passo para frente, a rede neural realiza o aprendizado com os dados de treinamentos e a cada saída gerada, no caso de classificações categóricas, indicam as probabilidades ou confianças dos possíveis rótulos. Essas probabilidades são comparadas com os rótulos verdadeiros, dessa maneira, a função de custo, calcula uma penalidade para qualquer desvio entre os dois rótulos. Em outro momento, na retro propagação, a derivada parcial da função de custo é responsável por indicar a direção que leva a função de custo a obter o menor valor possível, dessa forma, calculando novos pesos (HO; WOKEY, 2020).

A função de custo Entropia Cruzada Categórica (do inglês, *Categorical cross-entropy*) é amplamente utilizada em combinação com a função de ativação *Softmax* na camada de saída, para problemas de classificação categóricos. A Equação 12 é descrita pela seguinte forma:

$$J_{cce} = -\frac{1}{M} \sum_{k=1}^K \sum_{m=1}^M y_m^k \times \log (h_\theta(x_m, k)) \quad (12)$$

Onde,

- m é o número de exemplos de treinamentos;
- k é o número de classes;
- y_m^k é o alvo para o exemplo de treinamento m para a classe k ;
- x é o exemplo de treinamento de entrada m ;
- h_θ é o modelo com pesos da rede neural θ .

2.4.5.3 Otimizadores

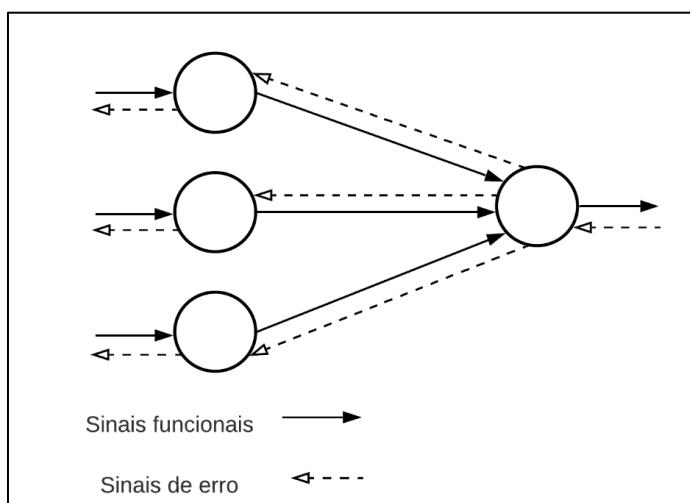
Um otimizador, determina como a rede será atualizada com base na função de custo. Ou seja, dita como a rede irá aprender. De maneira que, aprender significa descobrir um conjunto de valores para os pesos sinápticos para todas as camadas da rede, que irá conseguir mapear cada dado de entrada com o seu valor verdadeiro (CHOLLET, 2018).

Dessa forma, o otimizador utiliza o valor encontrado pela função de custo, como um sinal de *feedback* para ajustar os valores dos pesos, na direção que irá diminuir o valor da função de custo. Esse ajuste, é o trabalho do otimizador, de maneira que, sua implementação é chamada de algoritmo de retro propagação (do inglês, *backpropagation algorithm*) (CHOLLET, 2018).

Basicamente, esse algoritmo consiste em dois passos por meio das diferentes camadas da rede. Primeiro, um passo para frente é dado, passo de propagação, ou seja, os dados de entradas são aplicados nas camadas de entrada e esses dados passam camada a camada até a camada de saída e resultam em um valor. Durante o passo de propagação os pesos sinápticos e *biases* da rede são todos fixos (HAYKIN, 2008).

O segundo passo, consiste em dar um passo para trás, partindo da saída até a entrada. Dessa forma, os pesos sinápticos são ajustados de acordo o resultado do otimizador, que aponta quais pesos devem aumentar ou diminuir de acordo com desejo de toda a porção de dados utilizados pelo otimizador, como apresentado na Figura 12. Dessa forma, a cada iteração o resultado estará mais próximo do resultado esperado (HAYKIN, 2008).

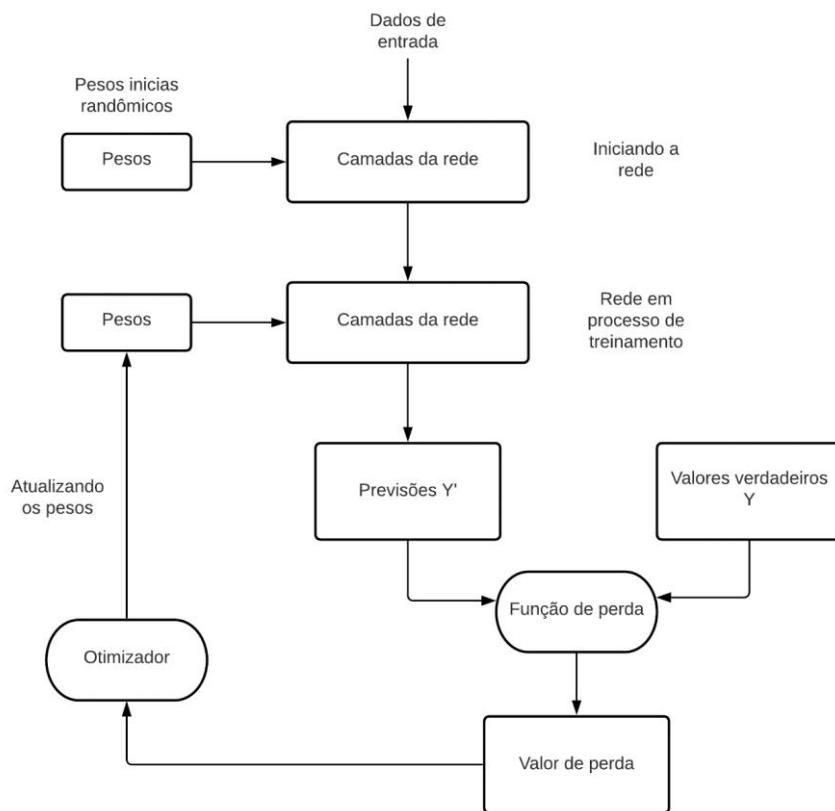
Figura 12. Retro propagação



Fonte: Adaptado de Haykin (2008).

Como apresentado na Figura 13, inicialmente, a rede começará com pesos randômicos, dessa forma, a sua função de custo terá um valor alto. Porém, a cada iteração esse valor irá diminuir pouco a pouco, há depender da taxa de aprendizado. Esse treinamento irá até que a minimização da função de custo seja alcançada, ou seja, o valor previsto será o valor esperado (valor verdadeiro) (CHOLLET, 2018).

Figura 13. Fluxograma de aprendizagem de uma rede neural



Fonte: Adaptado de Chollet (2018).

Um dos otimizadores mais moderno atualmente é o ADAM, proposto por Kingma e Ba (2015), ADAM é uma variação recente para o algoritmo do método do gradiente estocástico. A sua principal invocação e o cálculo não só das taxas de aprendizagem individuais para cada parâmetro, mas também do primeiro e segundo momento dos gradientes. O método é simples de implementar, é computacionalmente eficiente, tem poucos requisitos de memória. É voltado para problemas com grandes conjuntos de dados e/ou espaços de parâmetros de alta dimensão. O método combina as vantagens de dois métodos de otimização populares: a capacidade do AdaGrad de lidar com gradientes esparsos e a capacidade do RMSProp de lidar com objetivos não estacionários (KINGMA; BA, 2015).

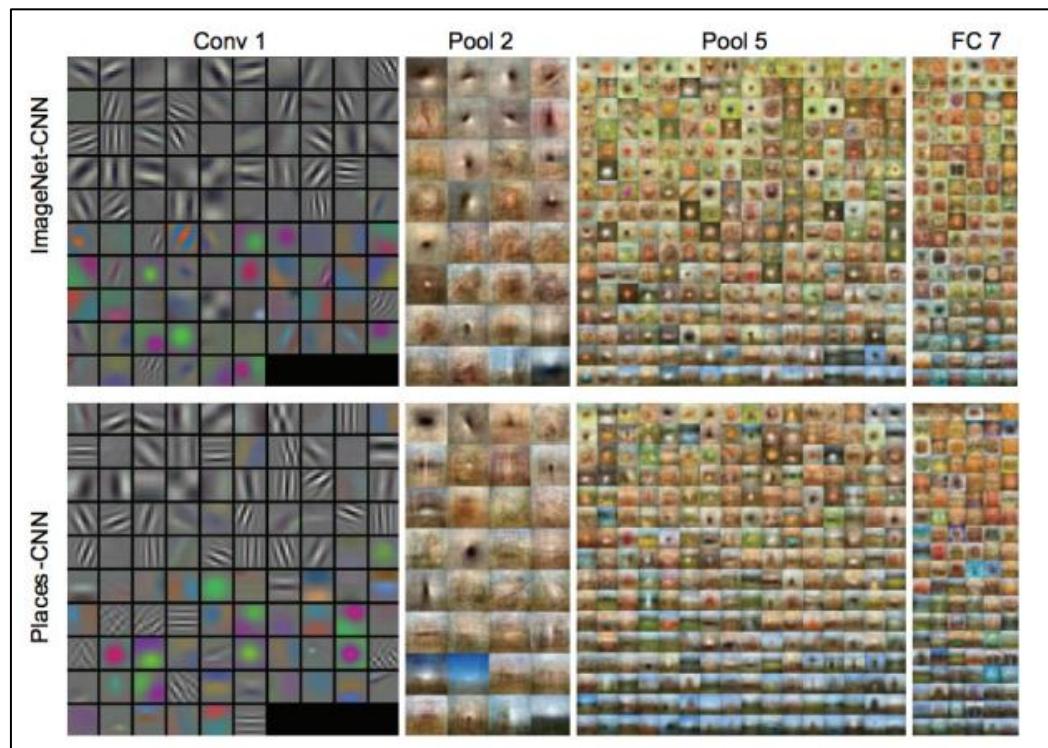
2.4.6 Transfer Learning

Os modelos pré-treinados passam por um processo de transferência de aprendizado (do inglês, *Transfer learning*, TL), de maneira que, a arquitetura passa por um pré-treinamento com um grande banco de dados, no caso desse trabalho, da ImageNet (RUSSAKOVSKY et al., 2015). Assim, os seus pesos aprendidos por esse pré-treino são ‘reciclados’ no processo de aprendizado dos dados escolhidos para o problema em questão (YOSINSKI et al., 2014).

Esse processo se chama afinação (do inglês, *fine-tuning*), de forma que, as camadas iniciais de uma CNN, normalmente aprendem conceitos muito gerais sobre as imagens e nas camadas finais da rede é onde estão os mapas de recursos mais específicos para determinado problema, como apresentado na Figura 14 (ZHOU et al., 2014).

As características passam de simples detectores de contornos e bordas para representar texturas e formas mais complexas de acordo com o avanço da imagem dentro da CNN. Dessa maneira, esse processo implica na redução do tempo e do poder computacional para o treinamento da arquitetura em questão.

Figura 14. Visualizações dos resultados dos filtros em diferentes níveis para duas arquiteturas de CNNs diferentes



Fonte: Zhou et al. (2014).

Efetivamente, além dos ganhos na eficiência, a menor quantidade de parâmetros para treinar e a redução dos riscos de *overfitting*, que é um importante problema no processo de treinamento. Os modelos pré-treinados de DCNN mais comumente utilizados para problemas que envolvam classificações de imagens são apresentados no Quadro 2.

De maneira que, a acurácia Top-1 e Top-5 refere-se ao desempenho dos modelos no conjunto de dados de validação ImageNet (RUSSAKOVSKY et al., 2015); A Profundidade se refere à profundidade topológica da rede. Isso inclui: camadas de ativação, camadas de normalização de lote (do inglês, *Batch Normalisation*) etc; E por fim, o Diferencial, que descreve a principal originalidade do modelo comparando-se com a arquitetura básica de uma CNN.

Quadro 2. Comparativo entre os modelos pré-treinados

Modelos	Tamanho	Top-1-Acc	Top-5-Acc	Parâmetros	Profundidade	Diferencial	Referência
Dense-Net121	33 MB	0.750	0.923	8,062,504	121	Cada camada obtém entradas adicionais de todas as camadas anteriores e passa em seus próprios mapas de características para as camadas.	HUANG et al., 2017
Dense-Net169	57 MB	0.762	0.932	14,307,880	169		
Dense-Net201	80 MB	0.773	0.936	20,242,984	201		
Res-Net50V2	98 MB	0.760	0.930	25,613,800	-	Cada camada se encaixa em um mapeamento residual. De maneira que, somasse o resíduo ao mapeamento subsequente.	HE et al., 2016
Res-Net101V2	171 MB	0.772	0.938	44,675,560	-		
Res-Net152V2	232 MB	0.780	0.942	60,380,648	-		

Xception	88 MB	0.790	0.945	22,910,480	126	Cada camada se encaixa em um mapeamento residual. De maneira que, somasse o resíduo ao mapeamento subsequente.	Chollet, 2017
InceptionV3	92 MB	0.779	0.937	23,851,784	159	Apresenta filtros com vários tamanhos operando no mesmo nível, tornando a arquitetura mais larga em vez de mais profunda.	SZE-GEDY et al., 2015
Inception-ResnetV2	215 MB	0.803	0.953	55,873,736	572	Introduz as conexões residuais que adicionam a saída da operação de convolução no módulo Inception de entrada.	SZE-GEDY et al., 2016
EfficientB0	29 MB	-	-	5,330,571	-	Um método de dimensionamento composto para dimensionar as dimensões da rede, sendo eles, profundidade, largura e resolução da rede.	TAN; LE, 2019
EfficientB1	31 MB	-	-	7,856,239	-		
EfficientB2	36 MB	-	-	9,177,569	-		
EfficientB3	48 MB	-	-	12,320,535	-		

Fonte: Adaptado de Chollet (2018).

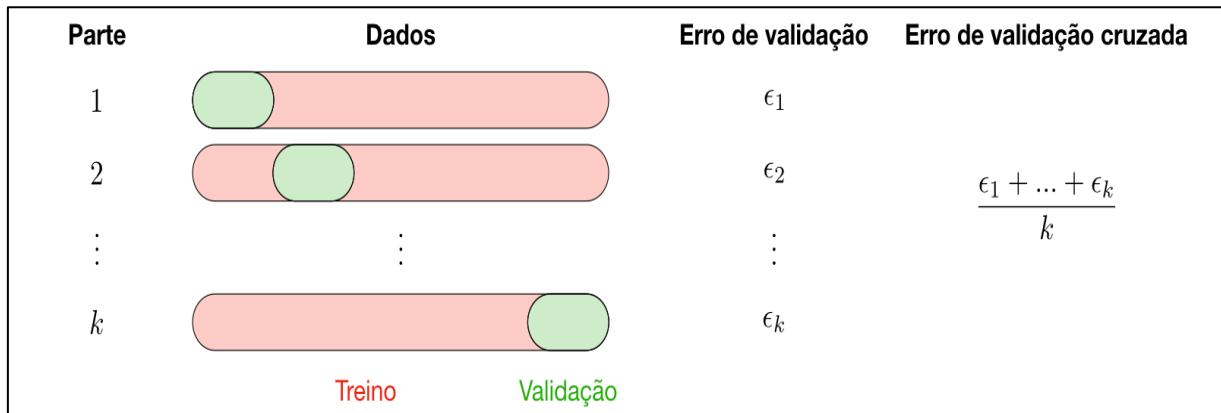
2.4.7 Métodos e algoritmos

Nessa subseção será apresentado o método de validação cruzada e dois algoritmos: Learning Rate Finder e o Cyclical Learning Rate. Os três serão utilizados na etapa de treinamento do modelo.

2.4.7.1 Validação cruzada de k-partes

A validação cruzada (do inglês, *Cross-Validation*, CV), é um método utilizado na seleção de modelos que não dependam muito do conjunto de treinamento inicial. A validação cruzada de k- partes (do inglês, *k-fold cross validation*) é uma técnica computacional intensiva, que usa todas as amostras disponíveis como amostras de treinamento e validação, demonstrada na Figura 15 (DUCHESNE; RÉMILLARD, 2005).

Figura 15. Demonstração do k-fold cross validation



Fonte: AMIDI & AMIDI (2018).

A técnica inicia-se com a divisão dos dados de treinamento em k partes (*folders*), onde esse k geralmente é 5 ou 10. Entre cada parte k, ocorre a divisão entre dados de treinamentos e de validação. Após isso, o modelo utiliza a k -1 partes para treino e a parte remanescente para validação, fazendo isso k vezes. Ou seja, a cada interação há um conjunto diferentes de treino e validação. Em cada uma das k vezes, o modelo é validado com uma parte diferente, calculando assim as métricas escolhidas pelo usuário, para avaliação do modelo. Ao final do processo, terão k medidas das métricas de avaliação escolhida. Por fim, O erro é então calculado sobre as k partes e é chamado erro de validação cruzada (do inglês, *cross-validation error*), ou seja, o resultado do nosso modelo vai ser a média dos valores de cada uma dessas partes. (AMIDI; AMIDI, 2018).

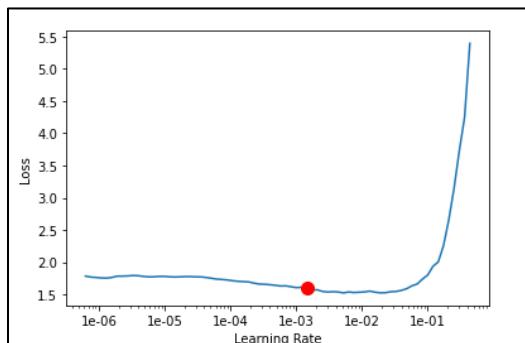
2.4.7.2 Learning Rate Finder

Proposto por Smith (2017), o *Learning Rate Finder* (LR), realiza uma busca pela melhor taxa de aprendizado (do inglês, *learning rate*), por meio, da realização de um teste de intervalo executado por algumas epoca (do inglês, *epoch*). De maneira que, esse teste consiste em realizar um treinamento com algumas taxas de aprendizado e ir aumentando a taxa após cada minilote (do ingles, *mini-batch*) até que o valor da perda comece a explodir.

A taxa de aprendizado, determina a variação dos pesos sinápticos da rede. Quanto menor for o parâmetro da taxa de aprendizagem, menor serão as variações, de uma iteração para a outra. Porém, isso causa uma aprendizagem mais lenta. Já para o caso de ter um parâmetro da taxa de aprendizagem alto, irá acelerar o processo de aprendizagem, de maneira, a obter grandes variações nos pesos sinápticos. Mas, isso pode resultar em uma rede instável (oscilatória) (HAYKIN, 2008).

Assim, a ideal inicial da execução do algoritmo é começar com um pequeno valor de taxa de aprendizado (como por exemplo, $1 \cdot 10^6$, $1 \cdot 10^5$) e ir aumentando a taxa após cada minilote até que a perda comece a explodir. Assim que a perda começar a explodir, o teste deve ser encerrado. Por fim, devesse plotar um gráfico da taxa de aprendizado e a perda, como demonstrado na Figura 16. Assim, devesse escolher a taxa de aprendizado de uma ordem abaixo da taxa de aprendizado, onde a perda é mínima (se a perda começar a baixar em 0,1, um bom valor para começar é 0,01). Este é o valor em que a perda ainda está diminuindo (SMITH, 2017).

Figura 16. Execução do LR. O valor encontra como melhor taxa de aprendizado foi de 1E-3, para um modelo de InceptionV3



Fonte: O autor.

Assim, para a atualização dos valores de taxas de aprendizado a cada minilote, devesse seguir as seguintes Equações (13, 14, 15 e 16), que demostram como encontrar a taxa de aprendizado ideal após o i-ésimo minilote:

$$\max_lr = \text{init_lr} \cdot t^n \quad (13)$$

$$q = \left(\frac{\max_lr}{\min_lr}\right)^{1/n} \quad (14)$$

$$lr_i = \text{init_lr} \cdot t^i \quad (15)$$

$$lr_i = \text{init_lr} \cdot \left(\frac{\max_lr}{\min_lr}\right)^{i/n} \quad (16)$$

Onde:

n é o número de iterações;

\max_lr é o valor máximo de taxa de aprendizado a ser utilizado. Normalmente se utiliza valores mais altos, como por exemplo, 10, 100. Observe que podemos não atingir esse valor lr durante o teste;

init_lr é o menor valor de taxa de aprendizado. teste irá iniciar a partir desse valor. Normalmente se utiliza um valor muito pequeno, como por exemplo, $1 \cdot 10^5$, $1 \cdot 10^6$.

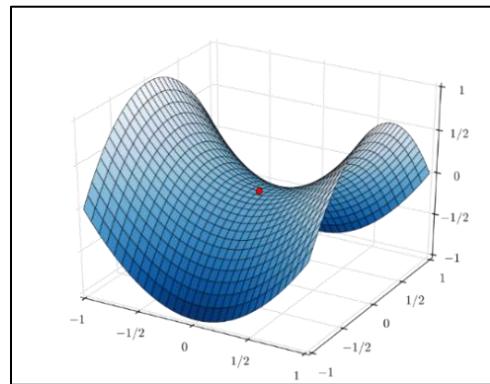
Assim, t , é o fator de aprendizado pelo qual será aumentado o valor de taxa de aprendizado após cada minilote. Por fim, uma vez que se encontrar o valor de taxa de aprendizado ideal, utilizasse esse valor para a realização do treinamento do modelo (SMITH, 2017).

2.4.7.3 Cyclical Learning Rate

Proposto pela mesma autora da seção anterior, Smith (2017), o *Cyclical Learning Rates (CLR)*, é um algoritmo que realiza a intercalação de um valor de taxa de aprendizado durante o treinamento entre o um limite inferior e o um limite superior. Convencionalmente, a taxa de aprendizado diminui à medida que a aprendizagem começa a convergir com o tempo. Dessa forma, surgiu a motivação por trás do *CLR* (SMITH, 2017).

Intuitivamente, é útil oscilar a taxa de aprendizado para um valor de taxa de aprendizado mais alto. Já que, com um maior valor de taxa de aprendizado *se* pode ajudar a sair dos pontos de selas, durante o treinamento. Se o ponto de sela for um platô elaborado (Figura 17) os valores de taxa de aprendizado mais baixos podem não conseguir obter gradiente fora do ponto de sela (SMITH, 2017).

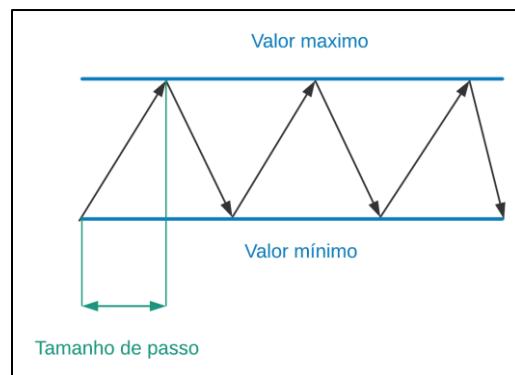
Figura 17. Ponto de sela



Fonte: Nicoguaro (2016).

O Ciclo é o número de iterações em oscilação entre os limites inferiores da taxa de aprendizado para o limite superior e de volta ao limite inferior. O ciclo pode não ter um limite durante uma época, mas na prática geralmente tem um tamanho de passo é a metade do ciclo. Portanto, tamanho de passo é o número de iterações em que a taxa de aprendizado vai de um limite para o outro, assim como, *max_lr* é a valor máximo que a taxa de aprendizado irá alcançar e *base_lr* é o valor mínimo, como demostrado na Figura 18. Esses dois limites podem ser decididos por meio do algoritmo descrito na seção anterior, *learning rate finder* (SMITH, 2017).

Figura 18. Abordagem de execução do algoritmo do CLR



Fonte: Adaptado de Smith (2017).

2.5 Trabalhos similares

2.5.1 Revisão Sistemática da Literatura

Para encontrar trabalhos relacionados ao tema de pesquisa, optou-se por realizar uma Revisão Sistemática da Literatura (RSL). Assim, para a execução da revisão seguiu-se as seguintes etapas: a delimitação da questão a ser tratada na revisão; a seleção das bases de dados bibliográficas para consulta e coleta de material; criação dos critérios de inclusão e exclusão; a

elaboração de estratégias para busca avançada; e por fim, a seleção dos trabalhos e análise dos resultados (KITCHENHAM; CHARTERS, 2007).

Adotou-se as diretrizes descritas em *Guidelines for performing Systematic Literature Reviews in Software Engineering* (KITCHENHAM; CHARTERS, 2007), assim como, as Diretrizes metodológicas: elaboração de revisão sistemática e metanálise de estudos de acurácia diagnóstica (SCTIE, 2014).

No Apêndice C, estão expostas todas as etapas do processo da revisão sistemática. Assim sendo, realizou-se uma seleção dentre as 52 publicações encontrado pela *string* de busca de acordo com os critérios de elegibilidade. Realizou-se uma leitura do texto completo para averiguar todos os critérios. Por fim, foram selecionadas 3 publicações.

2.5.2 Trabalhos selecionados

Com o apoio de instituições como a ISIC, que tem como objetivo agregar um amplo conjunto de dados de imagens dermatoscópicas e as disponibilizou ao público. Houve a quebra de uma barreira que impedia o avanço na aplicação de inteligência artificial (IA) em imagens aplicadas a área da saúde e mais especificamente a dermatologia (GUTMAN et al., 2016).

Dessa forma, por meio do ISIC, que atualmente tem um conjunto de dados composto por mais de 69445 imagens capturadas em diversos centros clínicos de diferentes países, por meio de uma variedade de dispositivos. A um aumento considerável em publicações relacionados a aplicação de IA na área de dermatologia e culminando para a adoção destes métodos no dia a dia dos serviços de saúde.

Desde 2016, o ISIC organiza o “ISIC: Skin Lesion Analysis Towards Melanoma Detection” que acontece anualmente e está em sua sexta edição no ano da apresentação desse trabalho. De maneira que, desafia cientistas globalmente de diversas áreas com problemas na segmentação de lesões de pele, detecção de padrões de diagnóstico clínico e classificação de lesões de pele. Levando a diversas descobertas e numerosas publicações anualmente. Como resultado, as publicações relacionados ao ISIC, foram aceitas em grupos de pesquisas como de fato, um padrão de referência.

Esse trabalho tem um foco na ISIC 2018, pois nesse desafio utilizou-se o HAM10000. Em virtude disso, para que ocorra uma comparação mais justa. Todos os trabalhos relacionados selecionados utilizam o HAM10000 com conjunto de dados de treinamento, validação e teste.

Lee, Jung e Won (2018), propuseram o WonDerm, uma pipeline que realiza um pré-processamento nas imagens e após a classificação das imagens das lesões de pele. O processo da pipeline inicia-se com o redimensionamento das imagens para 448x448 e um processo de remoção de ruídos, como por exemplo, o cabelo na pele dos pacientes. Seguido de um processo de *data augmentation* (conjunto de técnicas capazes de gerar novas imagens a partir de uma única imagem), constituído de rotações randômicas e flips verticais. Os modelos de classificação passaram por um processo de *fine-tuning* utilizando as imagens do HAM1000 após a segmentação. A arquitetura da DenseNet com 4 blocos densos, foi utilizada com modelo de classificação. Foram utilizados 4 modelos em *ensemble* para a classificação final, de maneira que, obteve-se 89,99% e 78,50 de acurácia balanceada dos conjuntos de dados de validação e teste, respectivamente.

Shen et al. (2021), abordam por outro ângulo o problema de classificação de lesões de pele. Levando em consideração ambientes com poucos recursos, que podem ser prejudicados por suas limitações em custo computacional, devido a demanda dos treinamentos dos treinamentos de modelos *ensembles*, por exemplo. Para superar essas limitações, propuseram uma estratégia de aumento de dados de baixo custo e alto desempenho que inclui dois estágios consecutivos: uma busca por diversas tecnicas de aumento de dados (do ingles, *data augmentation*, DA) e busca por diferentes DCNN. Inicialmente é feito uma divisão do HAM10000 em dois conjuntos: treinamento e teste. Em seguida, o conjunto de treinamento é dividido em dois sub-conjuntos: treinamento e validação. No primeiro estágio, a estratégia é realizar a busca pelas melhores combinação de tecnicas de DA e parâmetros sob o critério da acuracia balaceada (BACC) por meio do método de 5-Fold CV. As técnicas utilizadas são limitadas em 12 opções, denominadas por eles como LCA (Low-Cost-Augment). Essa busca é baseada no treinamento dos candidatos de modelos de CNN pré-treinadas, e por fim, é realizou-se uma avaliação em cada busca usando o subconjunto de validação BACC, onde é feita a triagem da melhor opção. Na segunda etapa, as CNN pré-treinadas são ajustadas com a melhor estratégia de DA com o conjunto de treinamento completo. Por fim, o modelo que obtiver o melhor valor de BACC no subconjunto de teste será o selecionado para realizar a classificação final. A CNN pré-treinada final utilizada por eles foi a EfficientNet-B2 que alcançou um valor de acurácia balanceada no conjunto de teste de 85,30%.

Yao et al. (2021), propõe uma nova abordagem de DA, porém, com foco na baixa quantidade de imagens e o desequilibrio presente na maioria dos problemas que envolvem a area da saúde. Para resolver essas duas questões, três etapas são necessárias. A primeira, é

adicionar camadas de DropOut e DropBlock nas RegNetYs para lidar com a redução do *overfitting*; A segunda etapa, consiste na aplicação de uma versão modificação do algoritmo RandAugment, focado em imagem dermatoscópicas, um algoritmo que realiza uma estratégia automática de DA. Essa etapa é responsável por lidar com a baixa quantidade de imagens, e por fim, na terceira etapa, propuseram uma versão modificada da Focal-Loss, uma função de custo, chamada Multi-weighted Focal Loss (MWFL), que lida com desbalanceamento. Com a junção dessas três etapas, alcançou-se um valor de acurácia balanceada de 86,40% no conjunto de dados de teste.

Dessa forma, as publicações são apresentadas no Quadro 3, de forma resumida, mencionando, se foi utilizado algum processo de aumento de dados, metodologia utilizada, o modelo final é um *ensemble* e os resultados alcançados (BACC – acurácia balanceada) para o diagnóstico de lesões de pele usando técnicas de *deep learning* em imagens dermatoscópicas.

Quadro 3. Comparação entre os trabalhos selecionados

Autor	Aumento de dados	Imagen De Entrada	Metodologia	Ensem-ble	BACC – Valida-ção	BACC - Teste
Lee, Jung e Won (2018)	Sim	224x224	Afinamento com Segmentação + 4 DenseNet	Sim	89,99%	78,50%
Shen et al. (2021)	Sim	224x224	LCA + EfficientNet-B2	Não	-	85,30%
Yao et al. (2021)	Sim	224x224	RandAugment modificado + MWFL + RegNetY-3.2G-Drop	Não	-	86,40%
Montagna (2021)	Sim	448x448	Cyclical Learning Rates + Learning Rate Finder + 6 modelos	Sim		87 %

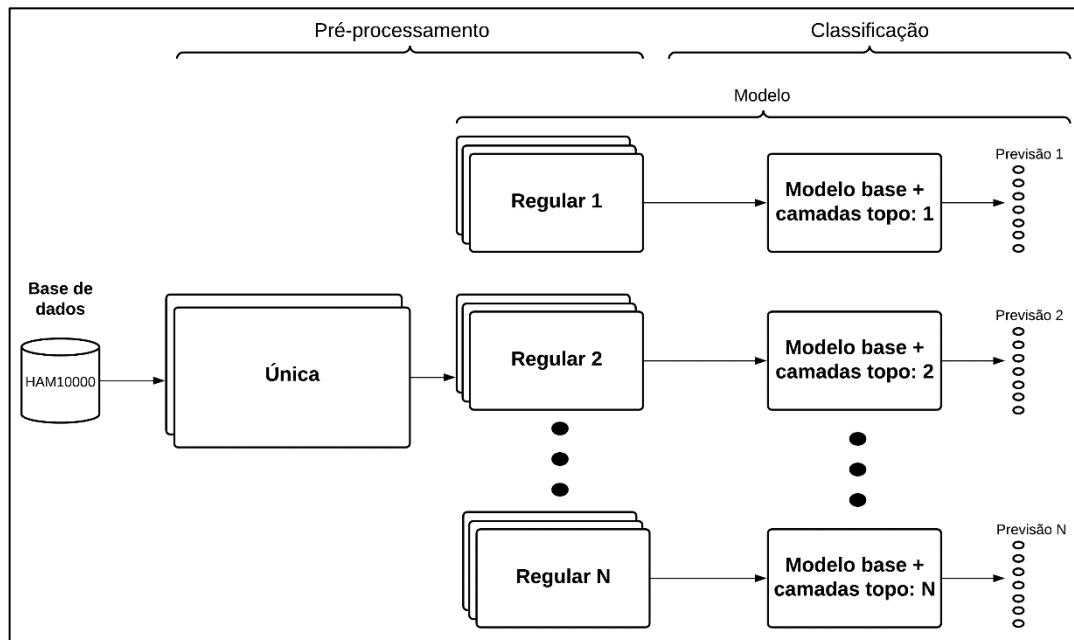
Fonte: O autor.

3 DESENVOLVIMENTO

Visando simplificar o entendimento, na Figura 19 é apresentado o fluxograma de desenvolvimento geral do modelo. Vale salientar, que foram desenvolvidos vários modelos para a realização de comparações e validações sobre algumas etapas para verificar a sua influência no desenvolvimento do modelo final.

Nesse fluxograma (Figura 19) é possível verificar que o processo se inicia com as imagens do HAM10000, seguido de uma etapa de pré-processamento, e por fim, a etapa final de classificação.

Figura 19. Fluxograma de desenvolvimento geral do modelo



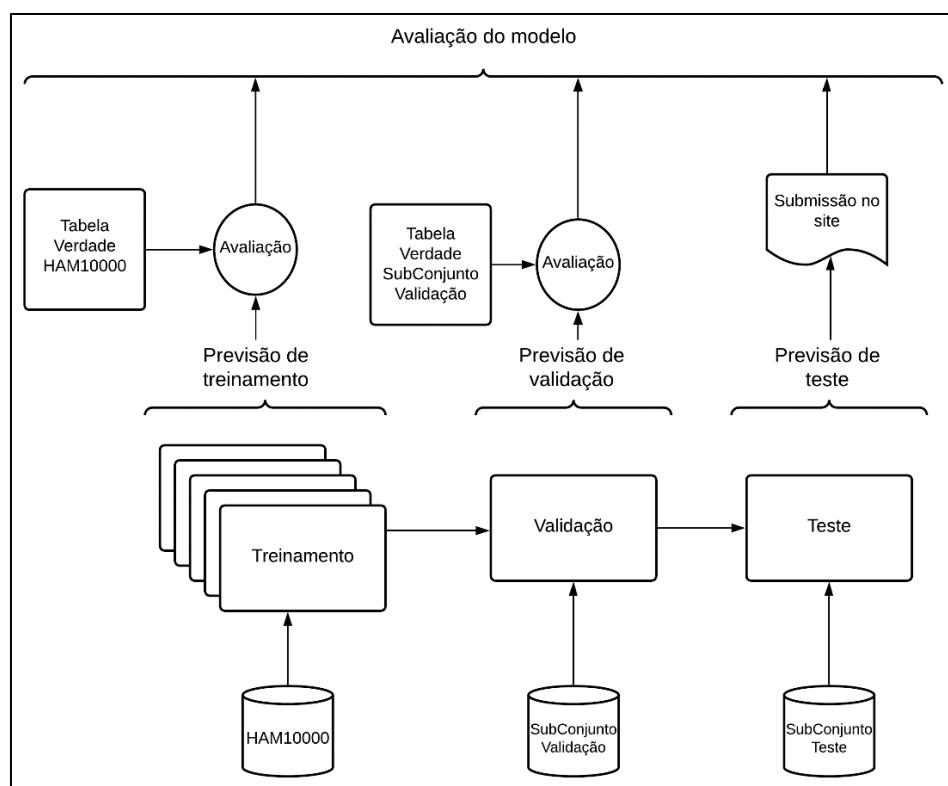
Fonte: O autor.

O processo inicia-se com a entrada de dados, nesse caso, o HAM10000. Em seguida, a etapa de pré-processamento, de maneira, que é dívida em duas partes. Sendo a primeira, a etapa Única, uma etapa que foi realizada apenas uma única vez no processo de treinamento e o resultado dessa etapa foi aplicado para todos os outros modelos que passaram pelo processo de treinamento. A segunda etapa do pré-processamento, denominada Regular, foi realizada a cada modelo treinado. Ou seja, regularmente a cada modelo que realizou o seu processo de treinamento. Por fim, a etapa de classificação, responsável por realizar a previsão. Que consta com o modelo que realizou o processo de previsão dos dados de entrada.

3.1 IMPLEMENTAÇÃO

Para exemplificar todo o processo de implementação, é apresentado na Figura 20, as etapas que serão explicadas mais em detalhes nas próximas seções. A implementação é dividida em 3 partes, sendo elas: o treinamento (que acontece 5 vezes, devido a utilização da validação cruzada de k-Partes) e a avaliação das previsões com a tabela verdade do HAM10000; a validação e a avaliação das previsões com a tabela verdade do subconjunto de validação; e por fim, o teste e a avaliação com a submissão no site.

Figura 20. Processos realizados durante a implementação



Fonte: O autor.

3.1.1 Ambiente

3.1.1.1 Linguagem de programação e bibliotecas

A linguagem de programação escolhida para esse trabalho foi o Python. Todos os experimentos, pré-processamento e arquiteturas foram avaliados e implementados empregando a biblioteca Tensorflow com a API Keras, como a biblioteca de DL.

Assim como, para a visualização dos resultados foram utilizadas as bibliotecas Matplotlib e Seaborn. Já para as manipulações de dados foram utilizadas as bibliotecas Numpy

e Pandas. Ademais, para avaliar as métricas de avaliação foram utilizadas as bibliotecas Sklearn e Mlxtend.

Além disso, a biblioteca Itertools foi utilizada para a manipulação de arquivos dentro dos ambientes de programação. Não apenas bibliotecas foram utilizadas, algumas implementações para os algoritmos Learning Rate Finder² e Cyclical Learning Rate³, encontradas no GitHub com licença livre, foram também utilizadas. Por fim, no Quadro 4, são demostradas as versões das bibliotecas citadas anteriormente utilizadas na implementação desse trabalho.

Quadro 4. Bibliotecas e versões utilizadas na data desse trabalho

Bibliotecas	Versão
TensorFlow	2.4.1
Keras	2.4.0
Matplotlib	3.3.3
Seaborn	0.11.1
Numpy	1.19.5
Pandas	1.1.5
Sklearn	0.24.1
Mlxtend	0.18.0
Itertools	3.9.4

Fonte: O autor.

3.1.1.2 Hardware

Todos os códigos foram implementados no Kaggle Code ou Google Colab Pro, ambientes de acesso livre para ser utilizados e que possuem as seguintes características apresentadas no Quadro 5.

Quadro 5. Especificações do ambiente Kaggle e Colab Pro

Dispositivo	Ambientes	Modelo	RAM (Gbs)	Cores	Memória do GPU (Gbs)	Perfomance (TFLOPS)
CPU	Kaggle	INTEL Xeon	16	2	-	-
	Colab	INTEL Xeon	25	2	-	-
GPU	Kaggle	NVIDIA P100	12	2	16	10.6
	Colab	NVIDIA P100	25	2	16	10.6
	Colab	NVIDIA T4	25	2	16	8.1

Fonte: O autor.

² Link para a implementação do LRF: https://github.com/benihime91/one_cycle_lr-tensorflow

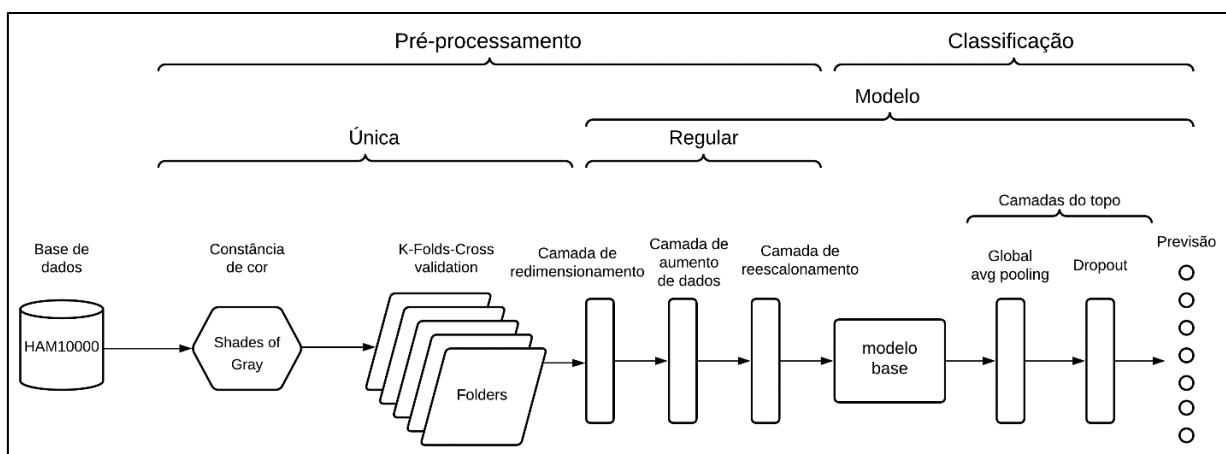
³ Link para a implementação do CLR: <https://github.com/tisu19021997/cyclical-scheduler>

Todos os experimentos foram desenvolvidos no modo GPU (*Graphics Processing Unit*) pois o tempo de processamento é muito menor ao treinamento em modo CPU (*Central Process Unit*) (WANG et al., 2019).

3.2 TREINAMENTO

Detalhando o fluxograma de desenvolvimento geral da Figura 19, a Figura 21 demonstra o fluxograma de desenvolvimento do modelo detalhado. De maneira, que cada processo será explicado nas seções abaixo.

Figura 21. Fluxograma de desenvolvimento do modelo geral detalhado



Fonte: O autor.

3.2.1 Base de dados: abordagens com o desbalanceamento das classes

O Quadro 6, apresenta a distribuição dos dados, como também, é possível verificar que a uma extensa quantidade de imagens dermatológicas de Nevo melanocítico sobre as outras categorias. Dessa forma, verifica-se que o HAM10000 tem um alto nível de desequilíbrio entre as sete categorias existentes.

Quadro 6. Distribuição dos dados

Nome	Categoria	Quantidade	Porcentagem
Melanoma	MEL	1113	11,1133 %
Nevo melanocítico	NV	6705	66,94957564 %
Carcinoma basocelular	BCC	514	5,132301548 %
Ceratoses actínicas e carcinoma espinocelular	AKIEC	327	3,265102346 %
Ceratose benigna	BKL	1099	10,97353969 %
Dermatofibroma	DF	115	1,148277584 %
Lesão vascular	VASC	142	1,41787319 %
TOTAL		10015	100 %

Fonte: O autor.

Dessa maneira, foi realizada a regulação com base em pesos em cada classe. Com o fim de regularizar as classes minoritária e incentivar o modelo a classificá-las corretamente, isto é, o modelo ao aprender, irá prestar mais atenção as classes minoritárias (HARRISON, 2020). Como também, foi levado em conta, a avaliação do modelo com métricas de avaliação que levem o desbalanceamento em consideração: acurácia balanceada, precisão, revocação e AUC (Mais detalhes podem ser vistos no Apêndice B).

Com a utilização da Equação 16, dado que o total de imagens presentes em um conjunto de dados de treinamento é 8912 (88.987 % do conjunto total do HAM10000), essa quantidade de imagens é devido ao método validação cruzada k-partes. Os resultados dos pesos calculados são apresentados no Quadro 7, de maneira, que as proporções entre as categorias foram mantidas se comparadas com o conjunto total do HAM 10000. Outro ponto, relevante para a definição dos pesos a serem utilizado no modelo, é a gravidade da lesão. Dessa forma, ocorreu um aumento no peso da categoria melanoma. Pois, se trata da categoria mais grave entre as sete. As demais sofreram arredondamentos para cima.

$$Peso_{Categoria} = \frac{\frac{total\ de\ imagens}{n^o\ imagens\ da\ categoria}}{2} \quad (16)$$

Quadro 7. Pesos associados a cada categoria no momento do treinamento do modelo

Categoria	Quantidade de exemplares	Peso calculado	Peso utilizado
MEL	1067	4.18	20
NV	5822	0.77	1
BCC	479	9.30	10
AKIEC	297	15	15
BKL	1011	4.41	5
DF	107	41.64	42
VASC	129	34.54	35

Fonte: O autor.

Por fim, outra importante questão, abordado por Tschandl et al. (2018), sobre o HAM é a presença de imagens únicas. Do total de imagens, 5.515 imagens são únicas e 4.500 imagens tem variantes. A distribuição das quantidades de variantes das imagens é apresentada no Quadro 8.

Quadro 8. Distribuição das variantes das imagens

Nome	Classes	1	2	3	4	5	6
Melanoma	MEL	230	556	300	16	5	6
Nevo melanocítico	NV	4415	1398	804	72	10	6
Carcinoma basocelular	BCC	175	236	99	4	15	0
Ceratoses actínicas e carcinoma espinocelular	AKIEC	151	114	54	8	0	0
Ceratose benigna	BKL	440	440	165	32	0	6
Dermatofibroma	DF	40	52	24	0	0	0
Lesão vascular	VASC	64	50	24	4	0	0
TOTAL		10015	5515	2846	1470	136	30
							18

Fonte: O autor.

Dessa forma, foi realizado uma distribuição dos dados, utilizando-se validação cruzada k-partes. Assim, as imagens foram separadas com base no Código de identificação da lesão, ou seja, garantindo que as imagens da mesma lesão não estejam em uma divisão de treinamento e validação. Essa separação foi feita com a utilização dos metadados fornecidos por Tschandl et al. (2018), um arquivo csv, contendo: o id da lesão, o id da imagem, o diagnóstico, o modo de diagnóstico, localização da lesão, a idade, e por fim, o sexo do paciente.

Para a realização da carga dos dados após os processos acima, foram utilizados os seguintes parâmetros: o tamanho do *batch*, a dimensão das imagens, modo do rotulo igual a categórico, embaralhado igual a verdadeiro e semente igual a 42. Os parâmetros que não foram especificados, serão especificados nas próximas seções com mais detalhes.

3.2.2 Pré-processamento

Para o processo de pré-processamento, como apresentado na Figura 22, iniciou-se com a aplicação do algoritmo de constância de cor *Shades of Gray* com a distância de Minkowski Normalizada igual a 6 para todo o conjunto de dados do HAM10000 de treinamento.

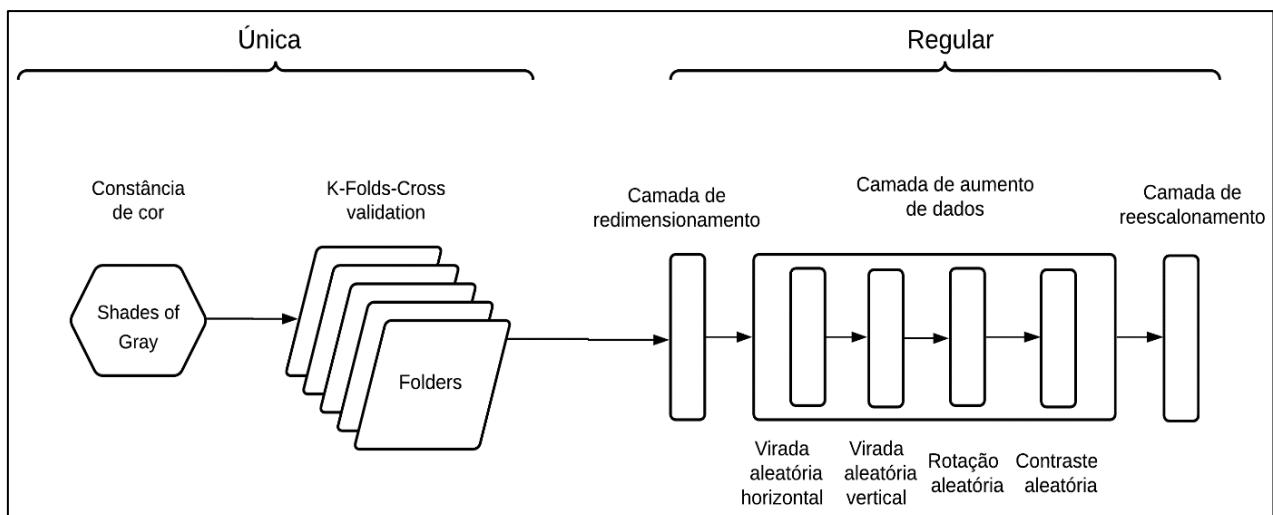
Após, aplicou o processo de validação cruzada k-partes (do inglês, *k-folds-cross-validation*, KFCV), que foi realizado por meio do arquivo de metadado disponibilizado pelos autores do HAM10000, levando em consideração a questão mencionada na seção 3.2.1 desse trabalho. Dessa forma, segue-se os seguintes passos para a aplicação do KFCV no conjunto de dados HAM10000:

- 1) Foi realizado a leitura do arquivo de metadado do HAM10000 em formato CSV, contendo os seguintes campos: *lesion_id*, *image_id*, *dx*, *dx_type*, *age*, *sex*, *location*.

lization; respectivamente: id da lesão, id da imagem, diagnóstico, tipo de diagnóstico, idade, sexo, localização da lesão. Dessa maneira, criando um *Dataframe*.

- 2) Os dados tabulares foram agrupados pelo campo ‘lesion_id’ e uma função ‘count’ foi aplicada para identificar quais lesões possuíram mais de uma imagem presente no conjunto de dados. Dessa forma, um novo campo ‘Tem_duplicatas’, foi criado dentro do *Dataframe*, de maneira, ‘Sim’ significa que contém mais de uma imagem dentro do conjunto de dados e ‘Não’ significa o inverso;
- 3) Assim, foi realizada a quintuplicação do campo ‘image_id’ com as seguintes nomenclaturas, respectivamente: fold_1, fold_2, fold_3, fold_4 e fold_5. Para serem utilizados no processo de aplicação de KFCV com os seguintes parâmetros: 5 divisões, estado aleatório = 43, embaralhamento = Falso. Levando a ter 8912 imagens (88.987 % do conjunto total do HAM10000) de treinamento e 1103 imagens de validação em cada *folder*;
- 4) Como última etapa, foram feitas as separações das imagens do conjunto de dados HAM10000 em pasta que continham uma pasta de treino e outra de validação, sendo realizada a divisão de acordo com o resultado obtido da aplicação do KFCV. Por fim, obter-se 5 pastas (5 *folders*), tendo duas pastas internas cada uma, treino e validação; E dentro das pastas treino ou validação, continha uma pasta para cada categoria presente no HAM10000.

Figura 22. Fluxograma do pré-processamento utilizado no treinamento do modelo



Fonte: O autor.

Iniciando a etapa Regular (Figura 22), que além do ponto da repetição a cada modelo, outro ponto, que diferencia essa etapa da Única. É o fato que são camadas que fazem parte do modelo, de maneira, que foram executas de forma síncrona com o resto das camadas, e se beneficiaram da aceleração da GPU. A primeira camada é a de redimensionamento, uma vez que, as imagens presentes no conjunto de dados HAM10000 tem o formato de 600x450 e por meio dessa camada, elas são redimensionadas para 448x448.

Seguindo para a camada de aumento dados, temos quatro técnicas que fazem parte dela. Sendo respectivamente as seguintes técnicas: Virada aleatória horizontal e vertical, rotação aleatória e contraste aleatório. Cada técnica tem seus respectivos parâmetros, sendo eles:

- Virada aleatória: Contém o parâmetro ‘*mode*’ que pode ser horizontal, vertical e horizontal ou vertical. Esta camada tem a função de virar aleatoriamente as imagens com base no atributo de modo, como apresentado na Figura 23. Foram realizadas uma camada para horizontal e outra para vertical;

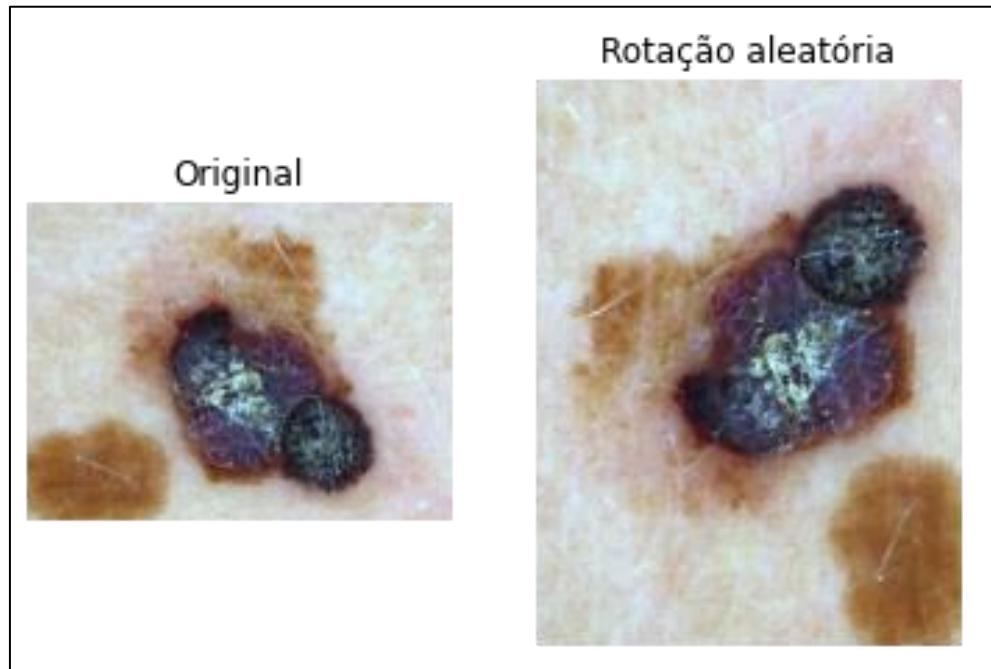
Figura 23. Exemplo da aplicação da virada aleatória horizontal e vertical



Fonte: o autor.

- Rotação aleatória: Contém o parâmetro ‘*factor*’ que é responsável por definir o intervalo de rotação aleatória. Esta camada tem a função de rotacionar as imagens com base no intervalo aleatório especificado, como apresentado na Figura 24. O fator utilizado foi 0.3, o que representa um intervalo inferior e superior $[-20\% * 2\pi, 2\pi * 20\%]$;

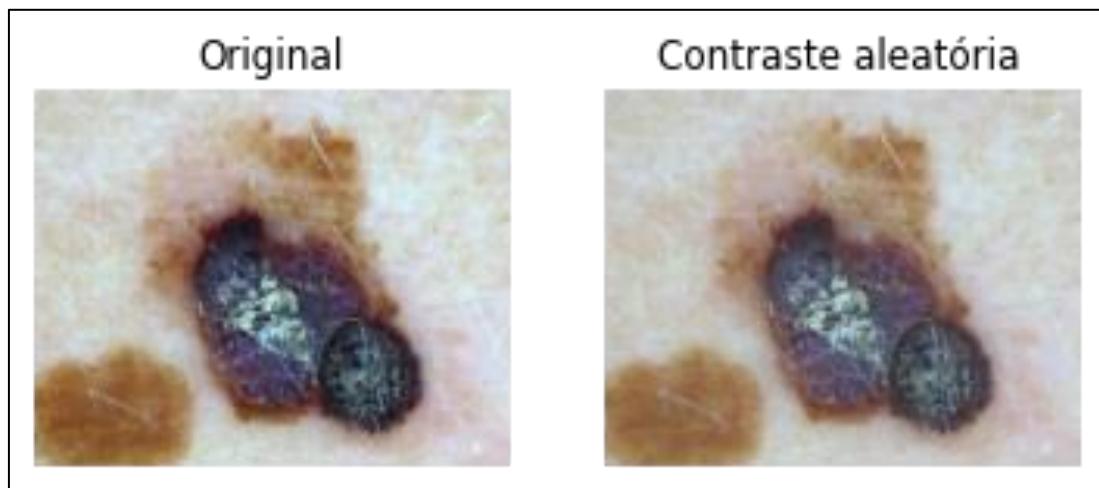
Figura 24. Exemplo da aplicação da rotação aleatória



Fonte: o autor.

- Contraste aleatória: Contém o parâmetro ‘*factor*’ que representa um limite inferior e superior para determinar o fator escolhido aleatoriamente para a aplicação do contraste na imagem, como apresentado na Figura 25. O intervalo para o fator utilizado foi [0.8, 1.1].

Figura 25. Exemplo da aplicação do contraste aleatório.



Fonte: o autor.

Por fim, temos a camada de reescalonamento, que realiza o processo de reescalar os valores de entrada de [0, 255] para [0, 1] ou [-1, 1], dependendo do modelo pré-treinado utilizado.

3.2.3 Classificação

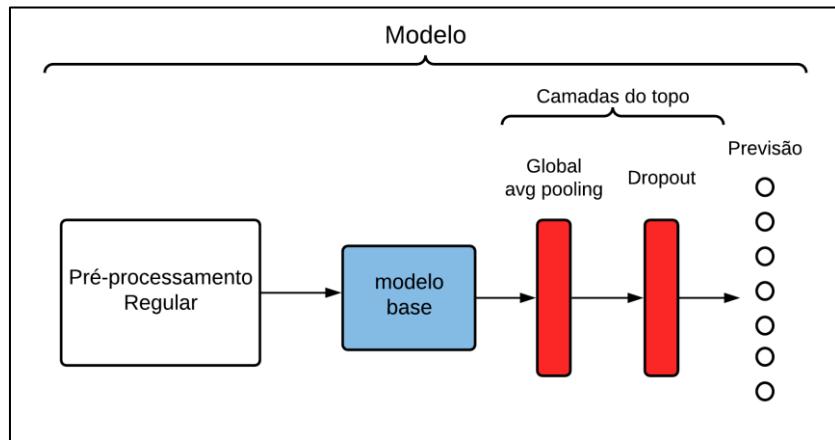
O modelo é formado por três partes, sendo a primeira as camadas de pré-processamento, seguido do modelo pré-treinado base e as camadas do topo que realizam o processo de classificação, como apresentado na Figura 26. A primeira camada, foi exposta na seção anterior.

Já a parte do modelo base, pode ser um dos dozes modelos pré-treinados. Os dozes modelos escolhidos são: DenseNet121, DenseNet169, DenseNet201, ResNet50V2, ResNet101V2, ResNet152V2, Xception, InceptionV3, InceptionResNetV2, EfficientNetB0, EfficientNetB1 e EfficientNetB2 (mais detalhes acerca dos modelos pré-treinados no Apêndice A). Durante o processo de treinamento, os modelos serão avaliados pelas seguintes métricas de avaliação: acurácia categórica, AUC, precisão e revocação (do inglês, Recall) (mais detalhes acerca das métricas de avaliação no Apêndice B).

A terceira parte, as camadas do topo, são formadas por uma camada *global average pooling*, que consiste em extrair por meio de uma janela, os valores médios presentes no mapa de característica em cada canal. Seguido por uma camada de *Dropout*, uma camada que ajuda a evitar que determinadas partes da rede neural tenham muita responsabilidade e consequentemente, possa ficar muito sensíveis a pequenas alterações, ou seja, tenham uma baixa generalização. Essa camada tem um hiper parâmetro que define uma probabilidade para certo neurônio, presente na camada, de ‘desligar’ em certo momento do processo de treinamento. Esse valor foi definido como 0.3 (30%) nos modelos implementados. Por fim, temos os sete neurônios para a previsão, com a função de ativação *Softmax*, que representam as sete categorias presentes no HAM10000.

O processo de treinamento do modelo acontece em dois momentos, o primeiro momento consiste em deixar todas as camadas do modelo base congeladas (seus pesos não serão treinados) e treinar somente as camadas presentes na camada do topo, como representado em azul (pesos congelados) e vermelho (pesos a serem treinados) na Figura 26. Vale ressaltar, que durante o processo de treinamento, também temos uma pequena parte, como exposta na seção 3.2.2 sobre o KFCV, do conjunto de dados completo do HAM10000 que ao final de cada época terá um processo de validação com o conjunto de validação da determinada pasta para verificar o estado atual do modelo.

Figura 26. Primeiro momento, fluxograma de classificação: Congelamento do modelo base e a realização do treino somente das camadas do topo



Fonte: O autor.

Para o início do treinamento, foram definidos: o otimizador ADAM, a função de perda Entropia Cruzada Categórica (do inglês, *Categorical Cross Entropy*), os pesos em cada categoria (definidos na seção 3.2.1), e por fim, os passos de treino e validação, que são definidos pela divisão do total de imagens presentes no conjunto de treinamento ou validação (subconjuntos de dados obtidos pela aplicação do método KFCV no HAM10000) pelo tamanho do minilote (do inglês, *mini-batch size*). O tamanho do minilote é um hiper parâmetro que define o número de amostras a serem utilizadas antes dos parâmetros do modelo serem atualizados. De maneira, que esse parâmetro foi definido como 16.

O tamanho do minilote influência diretamente no desempenho de treinamento. Com um tamanho maior, o modelo treinaria mais rápido a cada época, porém seria necessário um poder computacional muito maior do que o utilizado para o processamento paralelo das imagens. Além disso, outra desvantagem, é que mesmo com um poder computacional suficiente para lidar com lotes muito grandes, a qualidade do modelo pode degradar à medida que defizesse o tamanho dos lotes maiores e podem, em última análise, fazer com que o modelo seja incapaz de generalizar bem os dados não visto anteriormente.

Já o número de épocas (do inglês, *epoch*), é um hiper parâmetro que define o número de vezes que o algoritmo de aprendizado irá treinar o modelo em todo o conjunto de dados de treinamento. Uma época significa que cada amostra no conjunto de dados de treinamento teve a oportunidade de atualizar os parâmetros do modelo.

Uma época é composta por um ou mais lotes. Por exemplo, devesse pensar em um *loop for* sobre o número de épocas em que cada loop prossegue no conjunto de dados de treinamento.

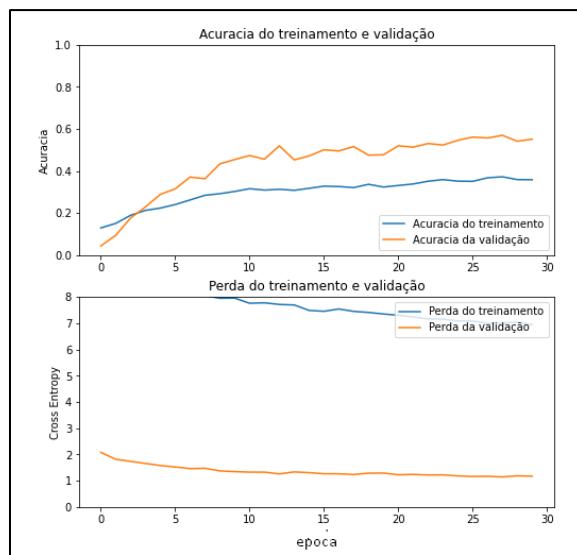
Dentro deste *for-loop* está outro *for-loop* aninhado que itera sobre cada lote de amostras, onde um lote tem o número de amostras de “tamanho de lote” especificado.

De forma, que esse valor para o primeiro momento do treinamento foram especificado em 30 épocas. Em suma, cada época durante o treinamento usará 557 minilote (*tamanho do conjunto de dados/tamanho do minilote*) para passar por todo o conjunto de dados e concluir uma época, ou seja, o numero de passos a serem obtidas do gerador antes de declarar o termino de uma época e iniciar a proxima.

O último hiper parâmetro a ser especificado é a taxa de aprendizado. Com a utilização do algoritmo LRF e a utilização do modelo, função de otimização, função de custo e o conjunto de dados de treino é possível ter uma sugestão de uma boa taxa de aprendizado para o treinamento do modelo. Essa taxa variou de acordo com o modelo base utilizado.

Após o fim das épocas do primeiro momento de treinamento, são analisados por meio de gráficos, a acurácia e perda de treinamento e validação sobre cada época, como demostrado na Figura 27. Para verificar se o modelo não se encaminhou para um *overfitting* (Mais detalhes podem ser vistos no Apêndice B.8.).

Figura 27. Exemplo de análise pós-treinamento do primeiro momento do modelo



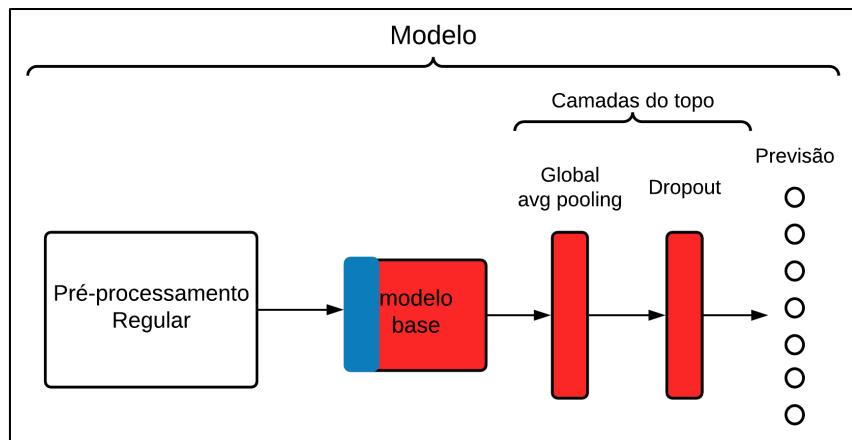
Fonte: O autor.

Dessa forma, o segundo momento de treinamento começa. As únicas diferenças entre o primeiro e segundo momento, consiste no afinamento (do inglês, *Fine-Tuning*) do modelo base, ou seja, o treinamento de apenas algumas camadas do modelo base e o restante congelado, não

treinável, preferencialmente as camadas inferiores (como exemplificado na seção 2.4.6), e a adição dos *callbacks*.

Vale ressaltar, que as camadas presentes na camada do topo passaram por um novo treinamento. Na Figura 28, é apresentado o modelo a ser treinado no segundo momento, em azul (pesos congelados), e em vermelho (pesos a serem treinados). A quantidade de camadas congeladas pode variar com o processo de afinamento, com exceção das camadas do topo.

Figura 28. Segundo momento, fluxograma de classificação: afinamento do modelo total



Fonte: O autor.

Durante o processo de treinamento do segundo momento, foram utilizados Callbacks. Os *Callbacks* são utilitários ou funções especiais que são executados durante o treinamento em determinados estágios do procedimento de treino dos modelos. Callbacks podem ajudar a prevenir *overfitting*, visualizar o progresso do treinamento, depurar seu código, salvar checkpoints, gerar logs etc. Durante o treinamento do segundo modelo foram utilizados três deles, sendo eles:

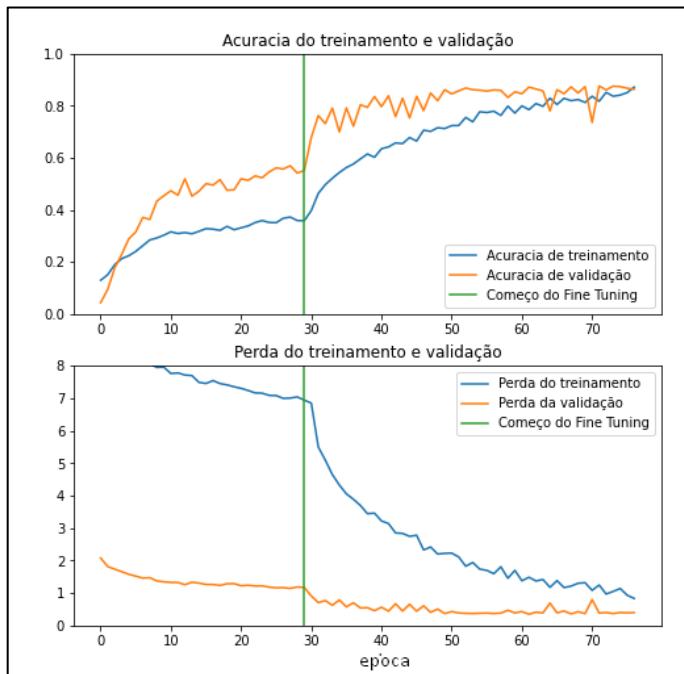
- O *callback* denominado *EarlyStopping*, tem a funcionalidade de interromper o procedimento de treinamento quando uma métrica monitorada, como por exemplo, perda ou acurácia do modelo, para de melhorar após um certo número de épocas. Esse *callback*, tem como função ajudar a reduzir o treinamento desnecessário do modelo quando ele não está melhorando, impedido assim, um *overfitting* no modelo (CHOLLET, 2018). Foi monitorada a métrica de perda de validação no modo mínimo, aguardando um período de 15 épocas.
- Já o *callback ModelCheckpoint*, tem a funcionalidade de salvar o modelo regularmente durante o treinamento. Isso é especialmente útil ao treinar modelos de

DL que levam muito tempo para treinar. Pois, dessa forma, o usuário não irá perder as informações importantes de treinamento em caso de falha do sistema. Esse *callback* monitora o treinamento e salva os pontos de verificação do modelo em intervalos regulares, com base nas métricas. O *ModelCheckpoint* salvará os pesos dos modelos como um arquivo de *checkpoint*, geralmente em um arquivo com extensão ‘.h5’ em seu sistema de armazenamento após cada época de sucesso ou após certas épocas que o usuário definiu explicitamente, por exemplo, melhor valor até agora de acurácia / perda (CHOLLET, 2018). Foi monitorado a métrica de avaliação AUC (mais detalhes no apêndice B.7.) de validação no modo máximo e salvando os melhores modelos.

- O último *callback*, o CLR, explicado na seção 2.4.7.3

Para o segundo momento do treinamento foram especificados mais 70 épocas, totalizando 100 épocas. Ou seja, o modelo treinou por mais 70 épocas, ou até o *callback EarlyStopping* ser ativado, além das 30 iniciadas no primeiro momento do treinamento. Por fim, após o treinamento do modelo no segundo momento, os mesmos processos de análise do modelo foram realizados, como apresentado na Figura 29.

Figura 29. Exemplo de análise pos-treinamento do segundo momento do modelo



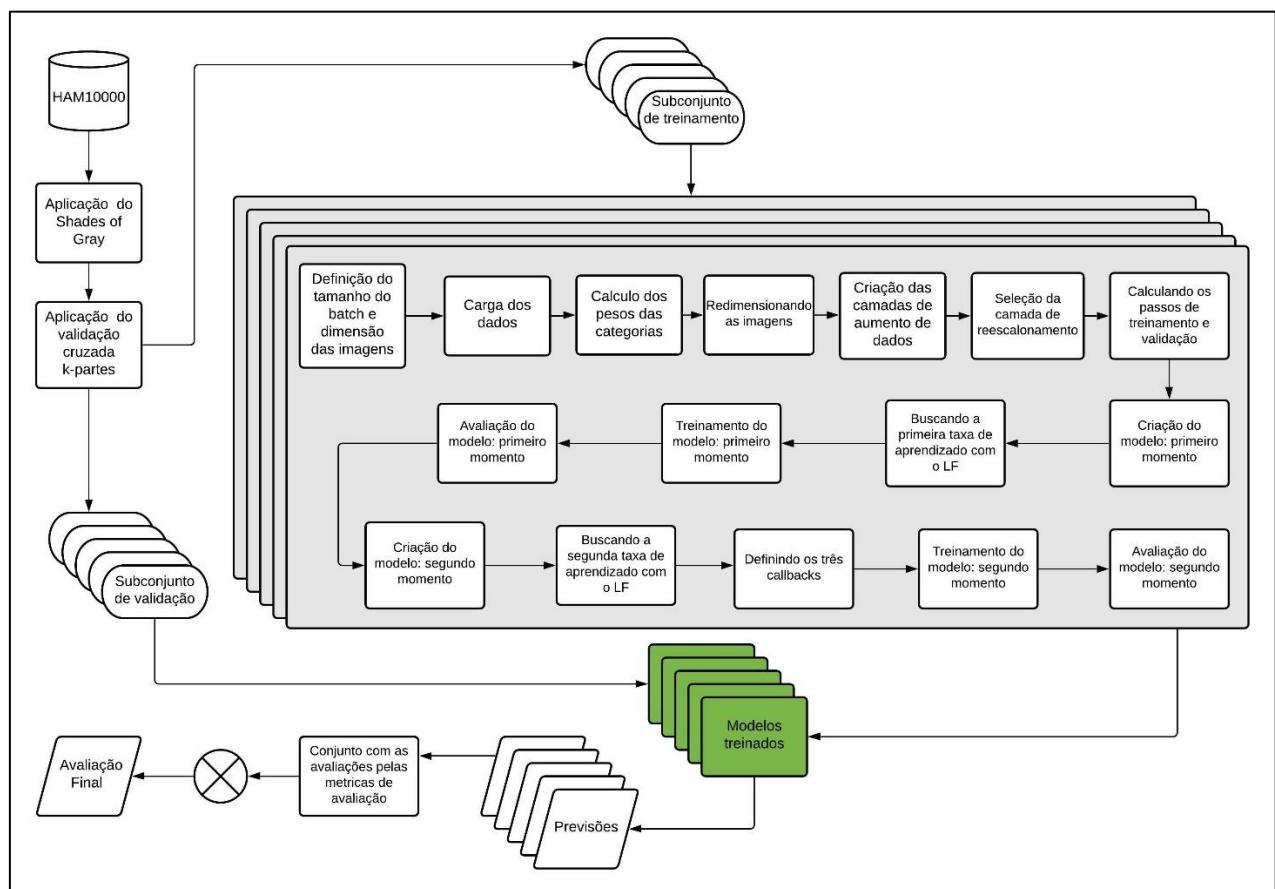
Fonte: O autor.

3.2.4 Resumo do treinamento

Todo o processo de treinamento explicado nas seções anteriores, é condensado na Figura 30. De maneira, que o processo inicia com o HAM10000 com a aplicação do algoritmo *Shades of Gray* e o método de validação cruzada em k-partes (criando as cinco pastas). Seguindo para o processo de criação do modelo, que tem como saída cinco modelos treinados que foram utilizados para gerar as cinco previsões com o subconjunto de validação.

Dessa forma, com as cinco previsões foi possível avaliar os modelos e realizar a média ponderada entre elas. Por fim, resultando na real avaliação do modelo, terminando assim, o método de validação cruzada em k-partes (KFCV). As métricas de avaliação utilizadas nesses processos finais do KFCV foi a acurácia balanceada.

Figura 30. Fluxograma resumindo o processo de treinamento



Fonte: O autor.

3.3 VALIDAÇÃO E TESTE

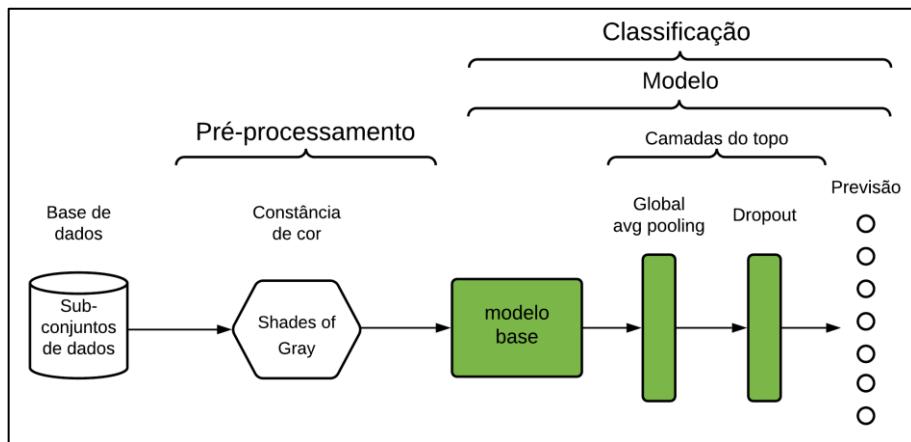
Os processos de validação e teste tiveram as mesmas etapas, diferenciando somente pelos subconjuntos de dados. O subconjunto de validação contém 193 imagens com uma tabela

verdade (tabela que contém os valores corretos das classificações), assim como, o conjunto de dados de treinamento.

Já para o subconjunto de teste, contém 1512 imagens e para verificar o número de classificações corretas e outras métricas de avaliação, no momento de escrita desse trabalho, devem ser submetidos no ISIC *Challenge* 2018.³⁴ as previsões resultantes do modelo. Ambos os subconjuntos são fornecidos pelo ISIC para o ISIC *Challenge* 2018.

Na Figura 31, é apresentado o fluxograma para a obtenção das previsões para os processos de validação ou teste do modelo. O modelo base e as camadas do topo em verde, significam que serão utilizados os pesos resultantes do treinamento com todo o conjunto de dados do HAM10000, ou seja, sem a utilização do método de validação cruzada k-partes.

Figura 31. Fluxograma de validação e teste com um subconjunto fornecido pelo ISIC 2018 Challenge



Fonte: O autor.

As métricas de avaliação utilizadas nesse processo de validação foram as seguintes: Curva Precisão-Revocação, Acurácia balanceada, Curva AUC-ROC e por fim, a Matriz de Confusão. Já para o de teste, somente a acurácia balanceada, pois, esse valor é obtido com a submissão de um arquivo tipo csv no site da ISIC⁴.

3.4 ENSEMBLE

Embora por uma questão de recurso computacional a junção de vários modelos em um, ocacione em um aumento no desempenho do modelo final, o contraponto surge com o aumento de hardware necessário, já que terá a necessidade de realizar o treinamento de vários modelos.

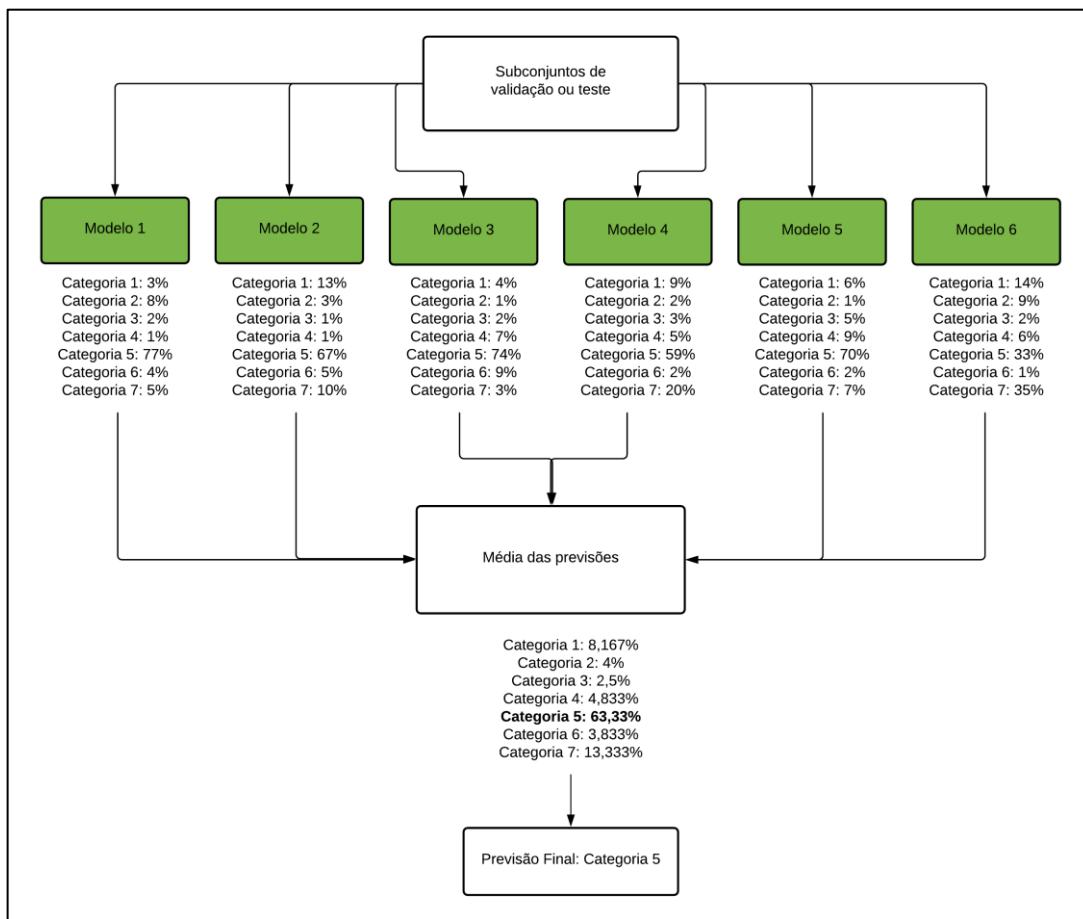
⁴ <https://challenge.isic-archive.com/task/51>

Essa abordagem é recomendada para o uso em competição e pesquisas, e não é recomendável para colocar em produção (MUSTAIN, 2020).

Uma técnica simples, mas eficiente para a realização do *Ensemble*, é calcular a média ponderada das previsões. Portanto, utiliza-se uma média das previsões de todos os modelos treinados e assim obtém-se a previsão final.

Dessa forma, cada modelo é treinado separadamente, seguido o fluxograma demostrado na Figura 32. Somente é adicionado um passo final de realizar a soma de todos os valores de previsões, que são probabilidades, e dividir pelo número de modelos somados para se obter um valor médio de previsão. Ou seja, a previsão do *ensemble*, como demostrado na Figura 32.

Figura 32. Método de aprendizado *ensemble*: Média das previsões



Fonte: O autor.

Assim, foi desenvolvido um modelo *ensemble* que continha os seis modelos que obtiveram as maiores acuráciais balanceadas. De maneira, que as métricas de avaliação utilizadas nesse processo *ensemble* foram as mesmas do processo de validação e teste, sendo elas: Curva Precisão-Revocação, Acurácia balanceada, Curva AUC-ROC e por fim, a Matriz de Confusão.

4 RESULTADOS E DISCUSSÕES

Os resultados do Quadro 9 são derivados do processo demostrado na Figura 30. Os valores presentes são dos doze modelos que passaram pelo processo de treino totalizando sessenta treinamentos. Os valores estão ordenados da maior para menor acurácia balanceada.

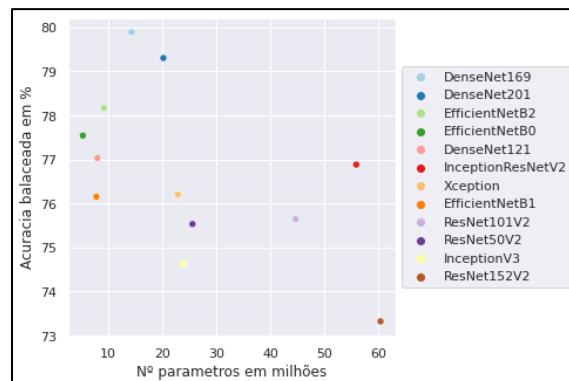
Quadro 9. Resultados do treinamento com KFCV

Modelo	Número de parâmetros	Média de acurácia balanceada (%)	Desvio Padrão (%)
DenseNet169	14,307,880	79,89361520	4,304442831
DenseNet201	20,242,984	79,30661720	2,590083357
EfficientNetB2	9,177,569	78,16106400	1,175951219
EfficientNetB0	5,330,571	77,53680920	3,653343465
DenseNet121	8,062,504	77,02600160	2,440388038
InceptionResNetV2	55,873,736	76,88256620	2,792707903
Xception	22,910,480	76,20396260	3,698807772
EfficientNetB1	7,856,239	76,15149720	2,639460572
ResNet101V2	44,707,176	75,63577280	2,031666377
ResNet50V2	25,613,800	75,53076200	1,109923223
InceptionV3	23,851,784	74,63173780	4,154442984
ResNet152V2	60,380,648	73,31784180	2,255964593

Fonte: O autor.

Como podemos verificar, os seis modelos que melhor se saíram no treinamento foram os seguintes: DenseNet169, DenseNet201, EfficientNetB2, EfficientNetB0, DenseNet121 e InceptionResNetV2. Na Figura 33, podemos verificar o fator desempenho, levando em consideração o número de parâmetros do modelo base e a acurácia balanceada alcançada. De maneira, que podemos observar que a maioria dos modelos que alcançaram as acuráncias balanceadas mais altas, não possuem um número tão alto de parâmetros. Demostrando, que o tamanho do modelo base não influência no resultado, mas sim, a forma ao qual a arquitetura está estruturada.

Figura 33. Representação em dispersão dos resultados obtidos com o treinamento com KFCV

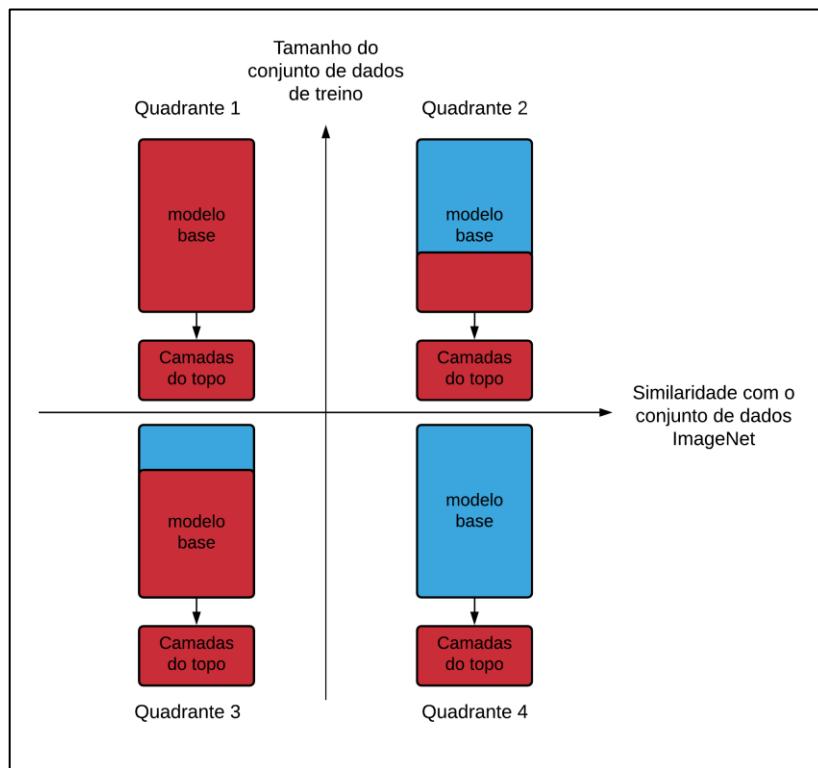


Fonte: O autor.

Alguns prontos que valem ser mencionados durante o segundo momento do treinamento. Dado um processo de afinamento do modelo, a quatro formas de proceder com o afinamento, levando em consideração que o conjunto de dados utilizados pelos modelos pré-treinados foi o ImageNet, sendo as quatro formas (como o apresentado na Figura 34):

- Quadrante 1: Um grande conjunto de dados e diferente do conjunto de dados ImageNet;
- Quadrante 2: Um grande conjunto de dados e com similaridade com o conjunto de dados ImageNet;
- Quadrante 3: Um pequeno conjunto de dados (aproximadamente seria ter menos de mil imagens por categoria) e diferente do conjunto de dados ImageNet.
- Quadrante 4: Um pequeno conjunto de dados e com similaridade com o conjunto de dados ImageNet;

Figura 34. Formas do afinamento



Fonte: Adaptado de Marcelino (2018).

Optou-se por utilizar a opção do quadrante 1 para o reaproveitamento do modelo pré-treinado. Devido à dificuldade de encontrar um equilíbrio entre o número de camadas treináveis e congeladas. Essa falta de equilíbrio causaria um sobreajuste do modelo ao conjunto de dados, ou também, caso fossem congeladas camadas demais, o modelo não aprenderia nada útil, um

subajuste. Ainda assim, outro ponto que levou a escolha do quadrante 1, é devido ao uso de doze modelos pre-treinados distintos, elevando a dificuldade de encontrar o equilíbrio para cada um dos doze.

Embora, o HAM10000 tenha algumas categorias com menos de mil imagens, o que o definiria com um conjunto de dados pequenos, o que resultaria pela escolha do quadrante 3 invéses do quadrante 1. Todavia, com o uso de técnicas de aumento de dados foi possível aumentar o número de exemplares das categorias que tinham menos de mil imagens, alcançando as definições do quadrante 1.

Apresentando no Quadro 10, os resultados obtidos com o processo da Figura 31. Podemos verificar que os seis modelos que alcançaram a acurácia balanceada mais alta foram: DenseNet201, DenseNet121, InceptionResNetV2, DenseNet169, EfficientNetB1 e ResNet101V2. Em comparação com os resultados encontrado no processo de treinamento, podemos verificar que dois modelos diferentes entraram no *top* seis: EfficientNetB1 e ResNet101V2. Isso será analisado comparando as matrizes de confusões desses oito modelos para entender essa diferença entre os modelos treinados com o método KFCV e com o conjunto de dados totais do HAM10000.

Quadro 10. Resultado do subconjunto de validação

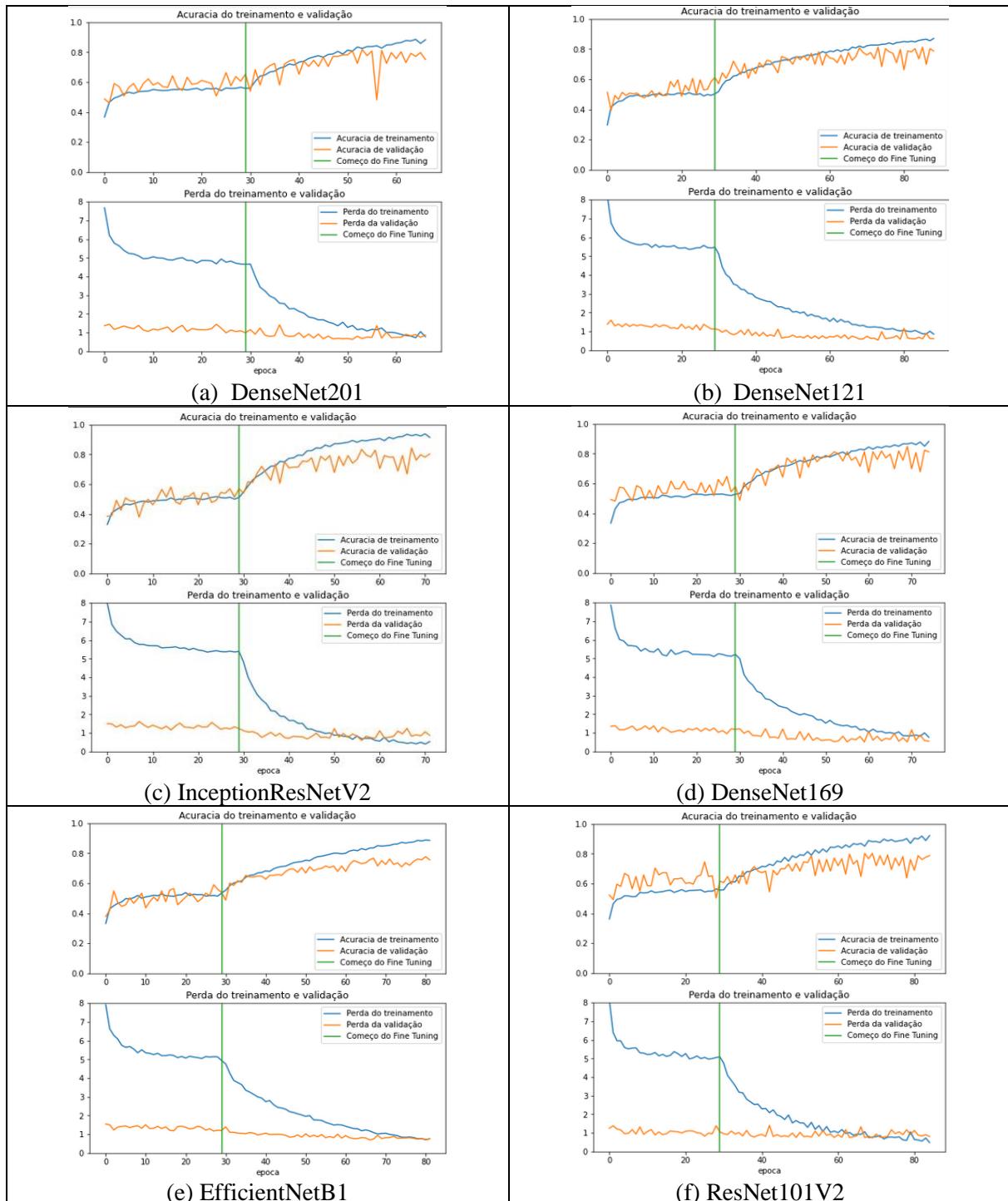
Modelo	Número de parâmetros	Média de acurácia balanceada
DenseNet201	20,242,984	87,693976
DenseNet121	8,062,504	84,771106
InceptionRes-NetV2	55,873,736	82,968158
DenseNet169	14,307,880	82,195348
EfficientNetB1	7,856,239	80,293113
ResNet101V2	44,675,560	79,504389
Xception	22,910,480	78,739536
ResNet50V2	25,613,800	78,334842
EfficientNetB2	9,177,569	77,707664
EfficientNetB0	5,330,571	77,106883
InceptionV3	23,851,784	76,876442
ResNet152V2	60,380,648	67,735870

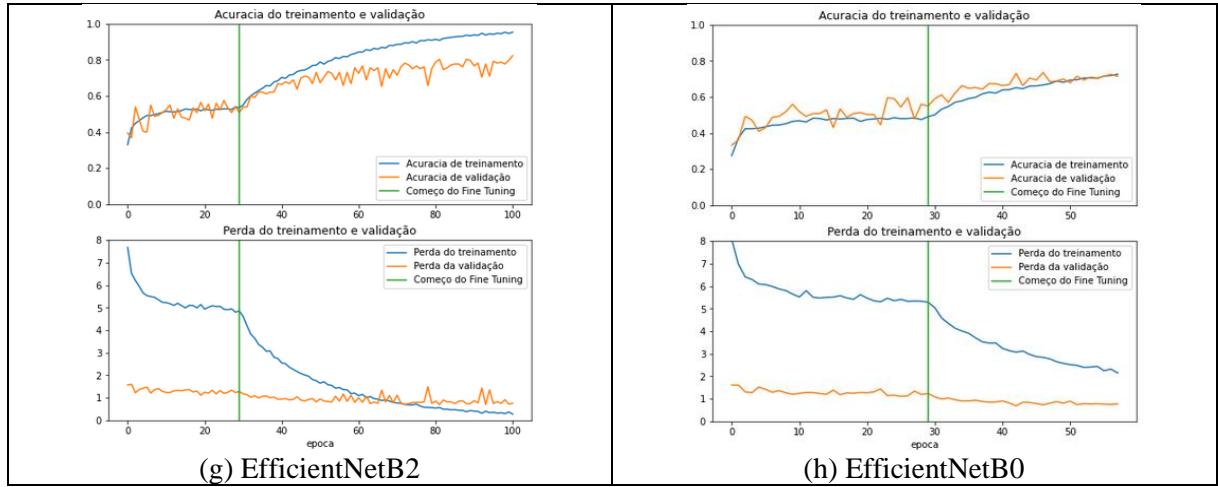
Fonte: O autor.

Primeiramente, analisando os modelos com os resultados pós-treinamento, podemos verificar na Figura 35 as curvas de acurácia e perda de treinamento e validação sobre cada época. Com exceção da EfficientNetB0 (Figura 35-h), todos outros modelos estão na medida (Apêndice

B.8.), levando em consideração a perda da regularização (perda associada a camada de Dropout da camada do topo, pois, ela é aplicada durante o treinamento e não durante a validação e teste). Nesse caso, a EfficientNetB0 poderia continuar o treinamento por mais algumas épocas para alcançar o estado ‘na medida’.

Figura 35. Comparativo entre as acurárias categoria e perdas de treinamento e validação sobre a época do processo abordado na seção 3.3



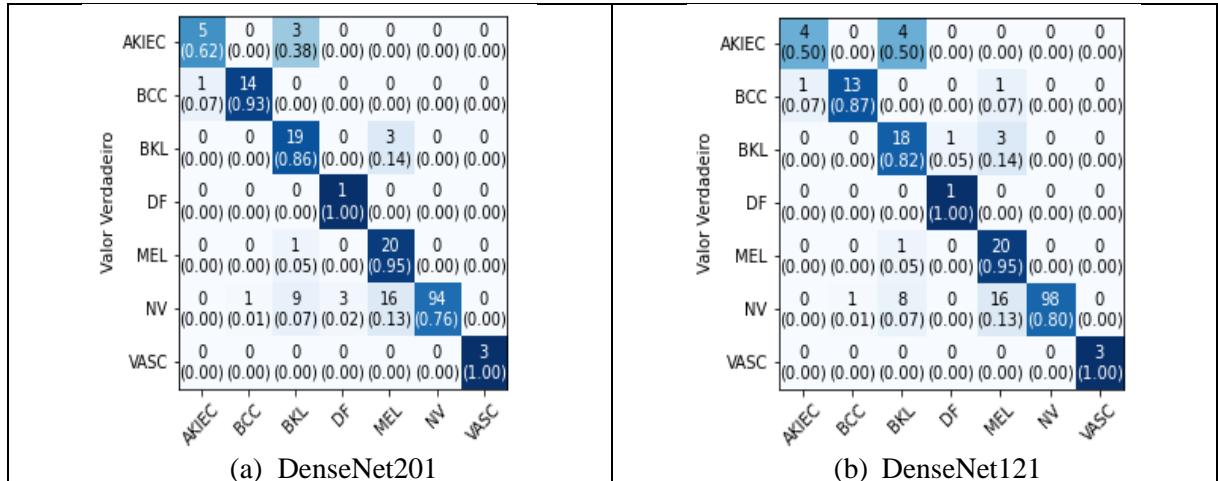


Fonte: o autor.

Explorando as matrizes de confusões dos oito modelos citados anteriormente, podemos verificar que entre as sete categorias, a que teve o menor número de acertos, foi a categoria AKIEC. Mesmo utilizando a entropia cruzada com pesos como a função de perda, que serviu para aliviar o desbalanceamento entre as categorias. De certa maneira, ajudou a superar as categorias majoritárias.

Entretanto, os pesos também podem ponderar uma categoria com uma taxa de verdadeiro positivos mais altas do que as outras categorias. Como ocorreu com DF e VASC. Ademais, as categorias BCC, BKL e NV performaram bem de acordo com os pesos estipulados. Por fim, a categoria MEL, que teve seu peso elevado acima das demais categoria, sendo, a categoria mais importante a ser classificada. Teve a sua taxa de verdadeiro positivo acima de 80% em todos os oito modelos.

Figura 36. Matrizes de confusão de oito modelos



		AKIEC	BCC	BKL	DF	MEL	NV	VASC		AKIEC	BCC	BKL	DF	MEL	NV	VASC	
Valor Verdadeiro	AKIEC	4 (0.50)	0 (0.00)	4 (0.50)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)		2 (0.25)	0 (0.00)	6 (0.75)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
	BCC	4 (0.27)	11 (0.73)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)		1 (0.07)	12 (0.80)	1 (0.07)	0 (0.00)	1 (0.07)	0 (0.00)	0 (0.00)	
	BKL	0 (0.00)	0 (0.00)	19 (0.86)	1 (0.05)	2 (0.09)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	21 (0.95)	0 (0.00)	1 (0.05)	0 (0.00)	0 (0.00)	
	DF	0 (0.00)	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	0 (1.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
	MEL	0 (0.00)	0 (0.00)	3 (0.14)	0 (0.00)	18 (0.86)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	1 (0.05)	0 (0.00)	20 (0.95)	0 (0.00)	0 (0.00)	
	NV	0 (0.00)	4 (0.03)	5 (0.04)	1 (0.01)	8 (0.07)	105 (0.85)	0 (0.00)		0 (0.00)	2 (0.02)	4 (0.03)	0 (0.00)	19 (0.15)	98 (0.80)	0 (0.00)	
	VASC	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)		0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)	
(c) InceptionResNetV2									(d) DenseNet169								
Valor Verdadeiro	AKIEC	3 (0.38)	0 (0.00)	5 (0.62)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)		3 (0.38)	0 (0.00)	5 (0.62)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
	BCC	2 (0.13)	12 (0.80)	0 (0.00)	1 (0.07)	0 (0.00)	0 (0.00)	0 (0.00)		1 (0.07)	11 (0.73)	1 (0.07)	0 (0.00)	2 (0.13)	0 (0.00)	0 (0.00)	
	BKL	0 (0.00)	0 (0.00)	19 (0.86)	0 (0.00)	3 (0.14)	0 (0.00)	0 (0.00)		1 (0.05)	0 (0.00)	18 (0.82)	0 (0.00)	2 (0.09)	1 (0.05)	0 (0.00)	
	DF	0 (0.00)	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	0 (1.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
	MEL	1 (0.05)	0 (0.00)	2 (0.10)	1 (0.05)	17 (0.81)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	4 (0.19)	0 (0.00)	17 (0.81)	0 (0.00)	0 (0.00)	
	NV	0 (0.00)	3 (0.02)	12 (0.10)	0 (0.00)	13 (0.11)	95 (0.77)	0 (0.00)		0 (0.00)	2 (0.02)	6 (0.05)	0 (0.00)	13 (0.11)	102 (0.83)	0 (0.00)	
	VASC	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)		0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)	
(e) EfficientNetB1									(f) ResNet101V2								
Valor Verdadeiro	AKIEC	2 (0.25)	0 (0.00)	5 (0.62)	0 (0.00)	1 (0.12)	0 (0.00)	0 (0.00)		2 (0.25)	0 (0.00)	2 (0.25)	1 (0.12)	3 (0.38)	0 (0.00)	0 (0.00)	
	BCC	3 (0.20)	9 (0.60)	0 (0.00)	0 (0.00)	3 (0.20)	0 (0.00)	0 (0.00)		0 (0.00)	12 (0.80)	0 (0.00)	2 (0.13)	1 (0.07)	0 (0.00)	0 (0.00)	
	BKL	0 (0.00)	0 (0.00)	19 (0.86)	0 (0.00)	2 (0.09)	1 (0.05)	0 (0.00)		0 (0.00)	0 (0.00)	16 (0.73)	0 (0.00)	5 (0.23)	1 (0.05)	0 (0.00)	
	DF	0 (0.00)	0 (0.00)	0 (0.00)	1 (1.00)	0 (0.00)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	0 (1.00)	1 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	
	MEL	0 (0.00)	0 (0.00)	2 (0.10)	0 (0.00)	19 (0.90)	0 (0.00)	0 (0.00)		0 (0.00)	0 (0.00)	1 (0.05)	1 (0.05)	19 (0.90)	0 (0.00)	0 (0.00)	
	NV	0 (0.00)	2 (0.02)	8 (0.07)	0 (0.00)	12 (0.10)	101 (0.82)	0 (0.00)		0 (0.00)	4 (0.03)	4 (0.03)	23 (0.19)	88 (0.72)	0 (0.00)		
	VASC	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)		0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	0 (0.00)	3 (1.00)	
(g) EfficientNetB2									(h) EfficientNetB0								

Fonte: o autor.

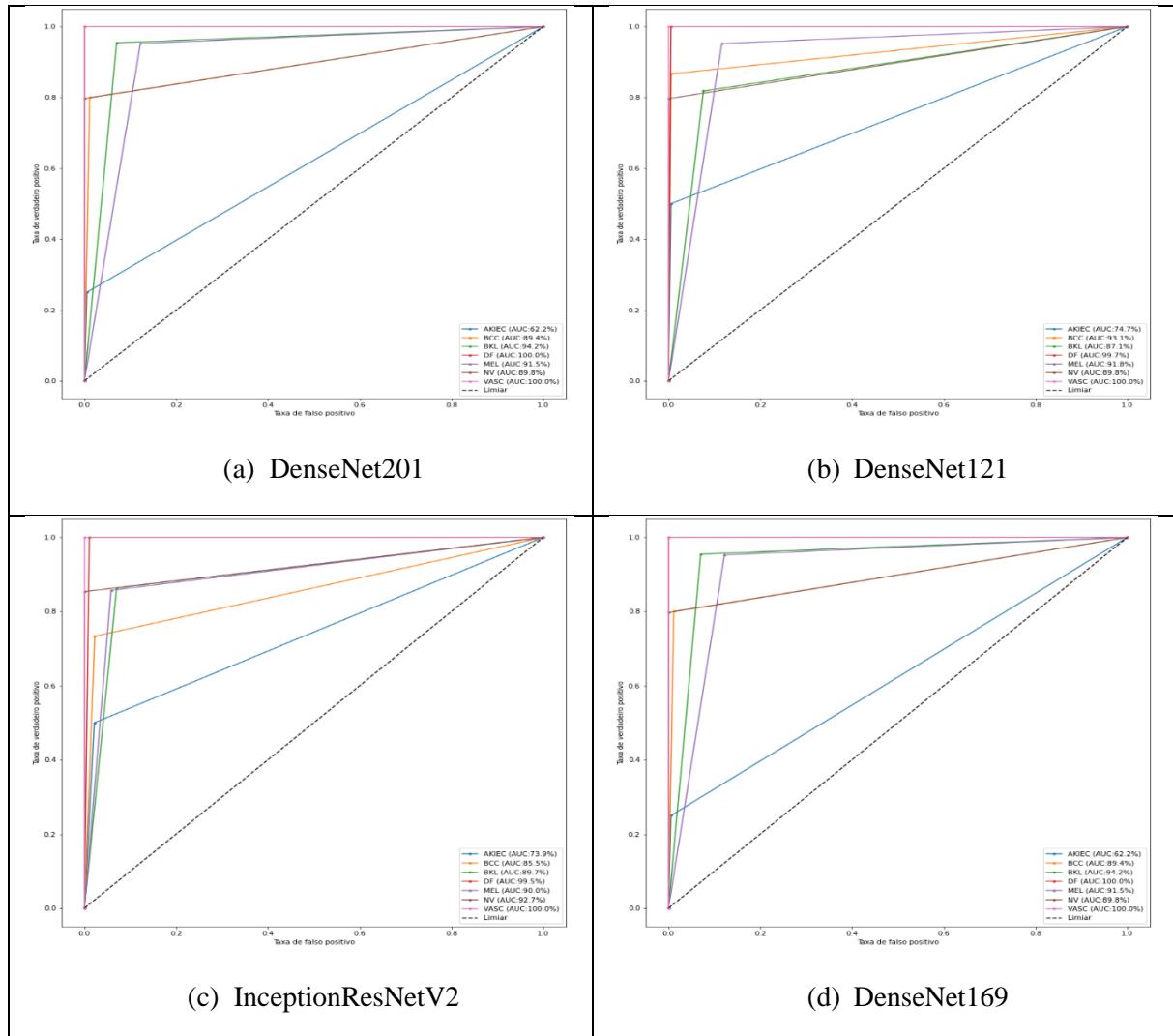
Ainda nas matrizes de confusões, podemos verificar que os dois modelos finais, EfficientNetB2 e EfficientNetB0, acabaram saindo do top seis no conjunto de validação, devido ao maior número de classificações errôneas em categorias que tinham menos exemplares. Sendo assim, punidas mais duramente pela acurácia balanceada, levando a esses dois modelos, a perda do top seis. Além disso, outro fator que pode ter influenciado essa diferença do treinamento

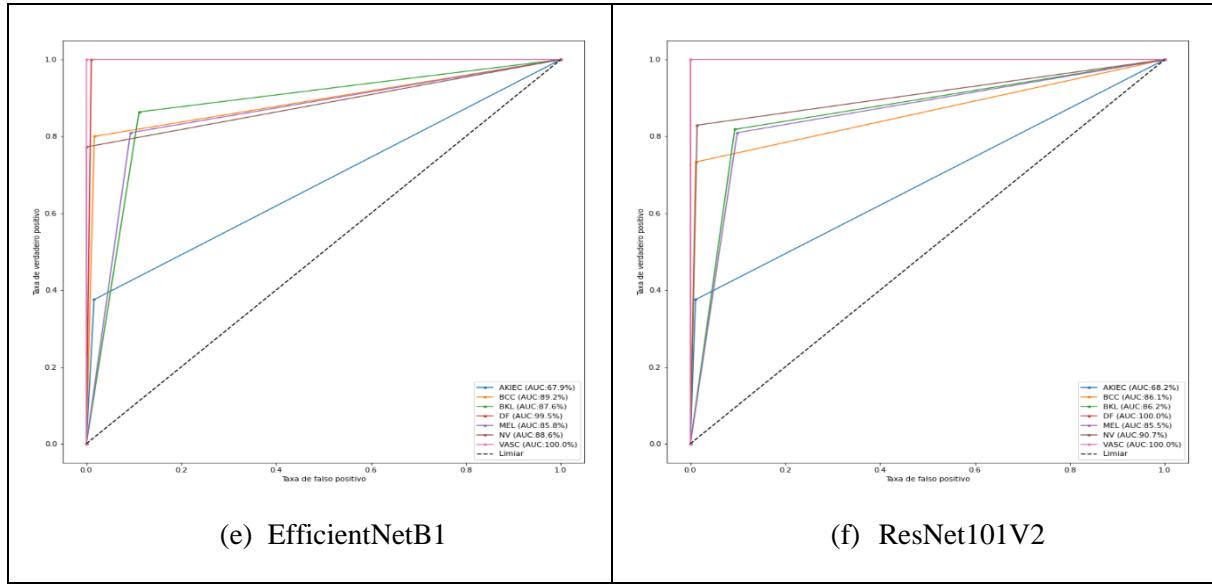
para validação, pode ter ocorrido devido à falta de mais exemplares de determinadas categorias na divisão das pastas na implementação do KFCV.

Avaliando agora, somente os modelos dentro do *top* seis do conjunto de validação por meio da curva AUC-ROC (Apêndice B.7.) apresentada na Figura 37. Essa métrica é útil para os dados do HAM10000, já que lida bem com dados desbalanceados, além disso, não é tendenciosa entre as classes majoritárias ou minoritárias.

Podemos observar, que dentre as sete categorias, AKIEC, como mencionada nas matrizes de confusões. Continua sendo a categoria com menos número de acertos, ou seja, com uma taxa menor de verdadeiros positivos. Apresentando um AUC acima de 60% para todos os seis modelos. Ademais, as outras seis categorias apresentaram um valor de AUC acima de 85%. O que acaba revelando, um alto valor de revocação, como será visto no próximo parágrafo.

Figura 37. Curvas ROC-AUC dos seis modelos com maioria acurácia balanceada no conjunto de validação



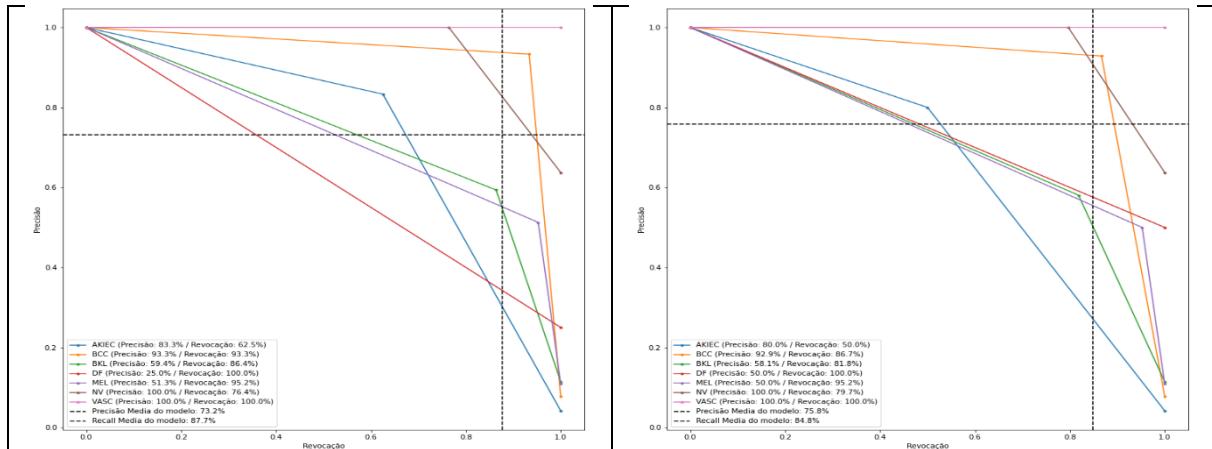


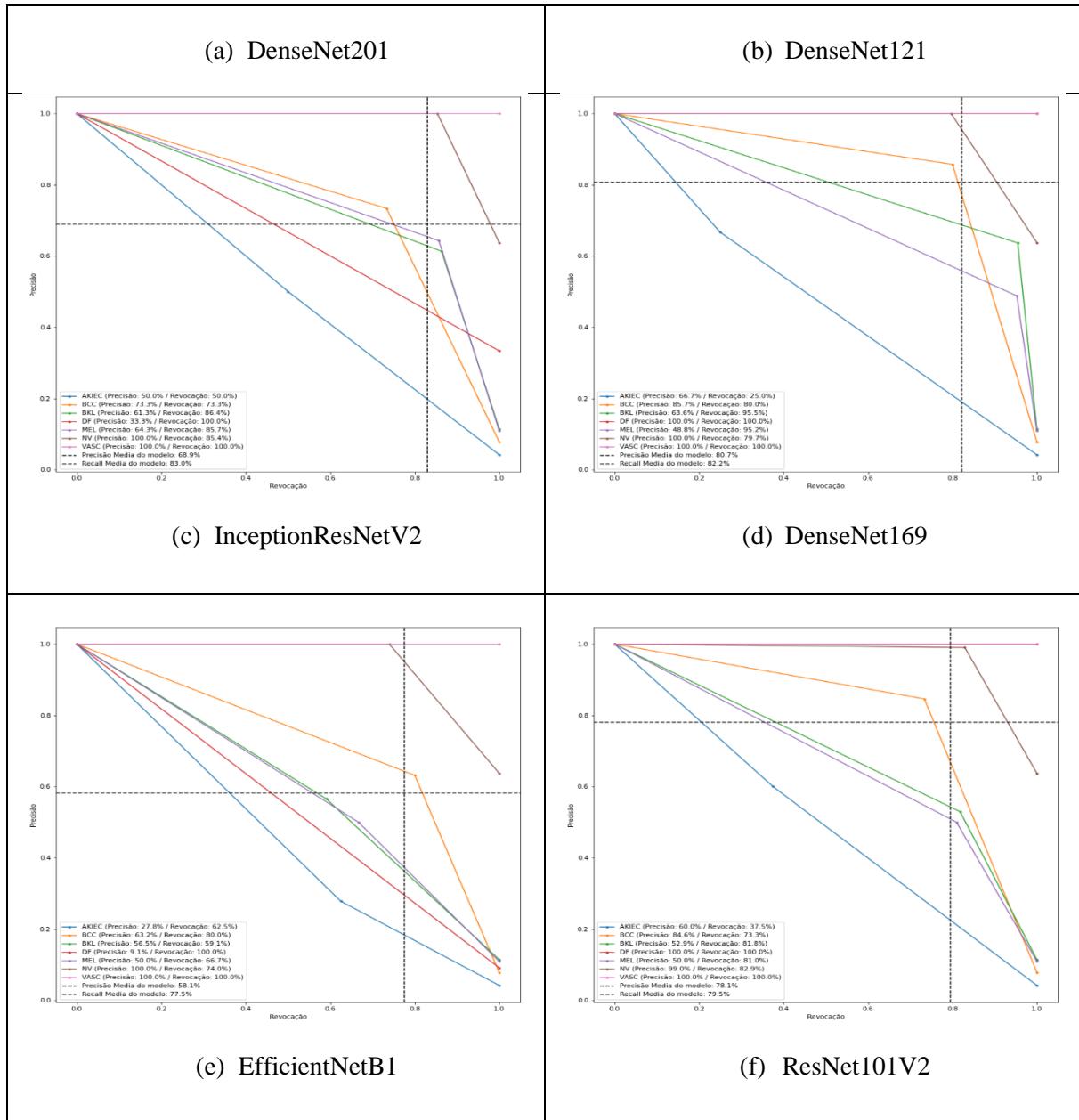
Fonte: o autor.

Verificando as curvas de precisão-revocação (Figura 38), é possível validar o que as Curvas de AUC-ROC apresentam. Todos os modelos apresentam um alto valor de revocação. No caso de modelos que auxiliam no diagnóstico, ter um *trade-off* (Apêndice B.3.) para a revocação é um ponto positivo.

Pois, devesse sempre minimizar os erros do tipo II (Apêndice B.1.), isto é, diagnosticar erroneamente algum paciente que não tenha uma lesão maligna e ele seguir para uma biopsia é muito melhor do que diagnosticar um paciente erroneamente que tenha uma lesão maligna e o diagnosticá-lo como uma lesão benigna e o paciente vir a falecer tempos depois. Dessa forma, minimizando as classificações falsas negativas.

Figura 38. Curva de Precisão-Revocação dos seis modelos com maioria acurácia balanceada no conjunto de validação





Fonte: o autor.

Aplicando o processo demostrado na Figura 31, dessa vez, para o subconjunto de teste. Uma vez geradas as previsões, os resultados eram salvos dentro de um arquivo CSV. De maneira, que cada linha continha o nome da imagem usada na previsão, seguida de suas probabilidades em cada categoria.

Assim, esses doze arquivos CSV foram submetidos no site da ISIC Challenge 2018 para obter-se o valor de acurácia balanceada. Pois, até a presente data de escrita desse trabalho, é a única forma de mensurar as previsões do subconjunto de teste. Já que não há uma tabela verdade disponível, como há para o conjunto de treinamento e o subconjunto de validação.

Consequentemente após a submissão dos doze arquivos, referente as previsões dos doze modelos. Foi possível obter os resultados apresentados no Quadro 11. Diante disso, podemos observar que o modelo DenseNet201, continua no topo como o melhor modelo de classificação. Surpreendentemente, o modelo ResNet101V2, apesar de ter uma acurácia balanceada com o subconjunto de validação o valor de 67,7%, apresentou um valor mais alto para o subconjunto de teste.

Quadro 11. Resultados do subconjunto de teste

Modelo	Acurácia balanceada (%)
DenseNet201	70,0400000
DenseNet121	69,9900000
DenseNet169	69,8000000
ResNet152V2	69,1000000
InceptionResNetV2	68,6000000
ResNet101V2	68,3000000
EfficientNetB2	67,2000000
Xception	67,1000000
EfficientNetB1	66,9000000
ResNet50V2	66,6000000
EfficientNetB0	65,5000000
InceptionV3	65,2000000

Fonte: o autor.

A diferença entre os valores encontrados no subconjunto de treinamento e teste é algo esperado, já que, os dados do subconjunto de teste nunca foram vistos pelo modelo. Dessa forma, é possível mensurar de forma correta o desempenho dos modelos. Outro ponto a ser considerado, é o tamanho dos dois subconjuntos, já que, o subconjunto de teste tem dez vezes mais imagens do que o subconjunto de validação, dessa forma, generalizando mais e tendo uma aproximação maior à um cenário real.

Por fim, a implementação do modelo ensemble. Seguindo o processo descrito e demonstrado na Seção 3.4. Para os subconjuntos de validação e teste, utilizando os seguintes modelos no ensemble: DenseNet121, DenseNet169, DenseNet201, InceptionResNetV2, EfficientNetB1 e ResNet101V2. Foi possível obter as acurárias balanceada respectivamente para os subconjuntos de validação e teste: 85,25% e 74,5%.

Para o subconjunto de validação, não houve uma melhora significativa, sendo que o modelo DenseNet201 alcançou sozinho 87,69 % de acurácia balanceada. Por tanto, essa abordagem em ensemble não foi eficiente para o subconjunto de validação. Já para o subconjunto de teste, ocorreu uma diferença significativa. Visto que, a maior acurácia balanceada resultante foi para o modelo DenseNet201 com o valor de 70,04% e o modelo em ensemble obteve um valor de 74,5%.

4.1 COMPARAÇÃO COM OS TRABALHOS RELACIONADOS

O Quadro 12 apresenta os resultados obtidos em comparação com os alcançados em trabalhos relacionados. Todos os trabalhos utilizam o HAM10000 como fonte de dados para o treinamento e os subconjuntos fornecidos pela ISIC 2018 para os processos de validação e teste dos modelos.

Quadro 12. Comparação com os trabalhos relacionados

Autor	Data Augmentation	Imagen De Entrada	Metodologia	Ensemble	BACC – Validação	BACC - Teste
Lee, Jung e Won (2018)	Sim	224x224	Fine-Tuning com Segmentação + 4 DenseNet	Sim	89,99%	78,50%
Shen et al. (2021)	Sim	224x224	LCA + EfficientNet-B2	Não	-	85,30%
Yao et al. (2021)	Sim	224x224	RandAugment modificado + MWFL + RegNetY-3.2G-Drop	Não	-	86,40%
Montagna (2021)	Sim	448x448	Cyclical Learning Rates + Learning Rate Finder + 6 modelos	Sim	85,25%	74,50 %

Fonte: o autor.

Como podemos ver pelos trabalhos relacionados, todos partem da premissa da utilização massiva de técnicas de aumento de dados. Alguns com processos e técnicas tradicionais e outros com mais modernas, como o utilizado pelo Yao et al. (2021). O presente trabalho, buscou uti-

lizar técnicas e processos tradicionais. além do fato, de utilizar apenas quatro técnicas de aumento de dados, buscando não modificar os aspectos fundamentais (como constante, recorte, nitidez, zoom etc) de cada imagem presente no conjunto de dados e nos subconjuntos.

Eventualmente, esse aspecto determinado para a construção do modelo pode ter tido uma alta influência no resultado, de maneira, que não foi possível alcançar o objetivo de 87% de acurácia balanceada no subconjunto de teste. Inquestionavelmente, tendo mais técnicas e até aplicado processos mais modernos de aumento de dados como o RandAugment (CUBUK et al., 2020), o modelo poderia alcançar um valor mais alto de acurácia balanceada, devido ao fato que, teria mais dados para treinar, e assim, generalizava mais.

Como também, aplicado processos mais complexos de processamento de imagens. Como por exemplo, a segmentação que poderia separar a lesão das demais partes da imagem, ou seja, extrair a região de interesse para a classificação; e a remoção dos pelos que podem influenciar em aspectos relacionados a borda das lesões.

Diante disso, os resultados alcançados foram satisfatórios. Tendo em vista que, os modelos foram construindo utilizando o dobro do tamanho de entrada dos dados, além, do fator das técnicas de treinamento utilizadas *Learning Rate Finder* e o *Cyclical Learning Rate*, que acabaram influenciado na eficiente de aprendizado dos modelos.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou uma metodologia para o auxílio no diagnóstico de lesões de pele, usando técnicas de aumentos de dados, algoritmos de processamento de imagem como o *Shades of Gray*, e por fim, doze modelos pré-treinados distintos para o desenvolvimento do modelo. O modelo *ensemble* alcançou um valor final de 74,50% de acurácia balanceada (acurácia que leva em consideração o desbalanceamento das classes).

Como contribuições, o fluxograma aplicado para os processos de treinamento, validação, teste e desenvolvimento dos modelos em *ensemble*. Assim como, a demonstração de uma forma de lidar com dados da área de saúde desbalanceados, como encontrar mais facilmente uma taxa de aprendizado para o treinamento do seu modelo e a importância da validação cruzada para o treinamento de novos modelos. Por fim, a demonstração que a família de modelos pré-treinado DenseNet demonstrou ótimo resultados para a classificação de lesões de pele.

Em conclusão, os resultados encontrados foram promissores levando em consideração a metodologia empregada. De maneira, que há muito a ser desenvolvido e validado. Considerando as diferentes possibilidades de construção de modelos na área de aprendizado profundo e a crescente necessidade da otimização de processos nas áreas médicas com a criação de sistemas de classificação, detecção, entre outras seria benéfico expandir a pesquisa considerando as seguintes propostas:

- Validar diferentes camadas do topo para a etapa de classificação, como por exemplo: SVM, Random Forest, Camadas Densas, CatBoost, GradientBoosting etc;
- Desenvolver uma pipeline de processamento de imagens mais robusta com processos como: segmentação, remoção de aspectos que causem problemas para a classificação da imagem. Além disso, aplicar algoritmos modernos de aumento de dados, como por exemplo, o RandAugment;
- Validar quais técnicas de aumento de dados influenciam mais em modelos de classificação de lesões de pele;

- Aplicar o afinamento das camadas para cada modelo base com a biblioteca Hyperactive, assim como, do número de neurônios para a camada do topo e para as camadas de regularização, por exemplo, a camada de *Dropout*;
- Validar outras técnicas de *ensemble* para os modelos, como por exemplo: *Bagging*, *Boosting*, *Stacking* e *Blending*;
- Avaliar outros modelos pré-treinados como por exemplo: SqueezeNet, GoogLeNet, ShuffleNet, ResNeXt, Wide ResNet, MNASNet, RegNet, PolyNet e SeNet;
- Validar e desenvolver uma abordagem que use em conjunto os metadados e as imagem dermatoscópicas. De maneira, que os metadados do paciente possam ajudar o modelo a classificar a imagem dermatoscópica de forma correta;
- Validar e desenvolver uma abordagem que realize um processo de classificação em um formato de duelo. Ou seja, múltiplas classificações binárias das categorias majoritárias até as categorias minoritárias. De maneira que, seriam agrupadas as categorias minoritárias em uma categoria e a outra categoria para a classificação binária seria a categoria majoritária dentro do conjunto de dados.

REFERÊNCIAS

- ACS, SOCIETY, A. C. *What Is Melanoma Skin Cancer?: What Is Melanoma?* 2017. Acessado em: 02 nov. 2020. Disponível em: <<https://www.cancer.org/cancer/melanoma-skin-cancer/about/what-is-melanoma.html>>.
- ADEGUN, A.; VIRIRI, S. Deep learning techniques for skin lesion analysis and melanoma-cancer detection: a survey of state-of-the-art. *Artificial Intelligence Review*, Springer, v. 54, n. 2, p. 811–841, 2021.
- AHMAD, J.; FARMAN, H.; JAN, Z. **Deep learning Learning: Convergence to Big Data Analytics.** USA. Editora: Now Publishers, 2019.
- AMAZON Machine Learning: Guia do desenvolvedor. Guia do desenvolvedor. 2021. Elaborado por: Amazon Web Services, Inc. and/or its affiliates. Disponível em: https://docs.aws.amazon.com/pt_br/machine-learning/latest/dg/machinelearning-dg.pdf#model-fit-underfitting-vs-overfitting. Acesso em: 23 jan. 2021.
- AMIDI, Afshine; AMIDI, Shervine. **Dicas e truques de aprendizado de máquina.** 2018. Traduzido por Fernando Santos. Revisado por Leticia Portella e Gabriel Fonseca. Disponível em: <https://stanford.edu/~shervine/l/pt/teaching/cs-229/dicas-truques-aprendizado-maquina>. Acesso em: 18 jan. 2021.
- ALVES, Gisely. **Viés, Variância e Regularização.** 2019. Disponível em: <https://dev.to/giselyalves13/vies-variancia-e-regularizacao-2me7>. Acesso em: 23 jan. 2021.
- ALI, M. S. *et al.* An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models. **Machine Learning with Applications**, [s. l.], v. 5, 2021. DOI 10.1016/j.mlwa.2021.100036. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S2666827021000177&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.
- BARATA, C.; CELEBI, M. E.; MARQUES, J. S. Improving dermoscopy image classification using color constancy. **IEEE Journal of Biomedical and Health Informatics**, v. 19, n. 3, p. 1146–1152, 2015.
- BRAY, F. *et al.* Global cancer statistics 2018: **GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries.** CA: a cancer journal for clinicians, Hoboken, v. 68, n. 6, p. 394-424, Nov. 2018.
- BRODERSEN, K. H. *et al.* The Balanced Accuracy and Its Posterior Distribution. **2010 20th International Conference on Pattern Recognition, Pattern Recognition (ICPR), 2010 20th International Conference on**, [s. l.], p. 3121–3124, 2010. DOI 10.1109/ICPR.2010.764. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsee&AN=edsee.5597285&lang=pt-br&site=eds-live&scope=site>. Acesso em: 24 jan. 2021.
- BOLKER, B.M. 2008. **Ecological Models and Data in R.** Princeton and Oxford: Princeton UniversityPress. 508 p.

CASTRO, Cristiano Leite de; BRAGA, Antônio Pádua. Aprendizado supervisionado com conjuntos de dados desbalanceados. **Sba Controle & Automação**, Campinas , v. 22, n. 5, p. 441-466, Oct. 2011 . Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-17592011000500002&lng=en&nrm=iso>. access on 23 Jan. 2021. <http://dx.doi.org/10.1590/S0103-17592011000500002>.

CHOLLET, François. **Deep learning with python**. Shelter Island: Manning Publications Co., 2018. 386 p. ISBN: 9781617294433.

CHOLLET, F. Xception: Deep Learning with Depthwise Separable Convolutions. **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ; ISBN 9781538604571**, [s. l.], 2017. DOI 10.1109/cvpr.2017.195. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.5CB77718&lang=pt-br&site=eds-live&scope=site>. Acesso em: 11 mar. 2021.

CFM, CONSELHO FEDERAL DE MEDICINA; CREMESP, CONSELHO REGIONAL DE MEDICINA DO ESTADO DE SÃO PAULO. **Concentração de Médicos no Brasil em 2020**. São Paulo, 2020. Disponível em: https://portal.cfm.org.br/images/stories/pdf/estudo_demografia_junho.pdf.

CONNOR SHORTEN; TAGHI M. KHOSHGOFTAAR. A survey on Image Data Augmentation for Deep Learning. **Journal of Big Data**, [s. l.], v. 6, n. 1, p. 1–48, 2019. DOI 10.1186/s40537-019-0197-0. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsdoj&AN=edsdoj.677e4d78a94b48aa897a5c41858b1562&lang=pt-br&site=eds-live&scope=site>. Acesso em: 23 jan. 2021.

CUBUK, E. D. *et al.* Randaugment: Practical automated data augmentation with a reduced search space. **2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Computer Vision and Pattern Recognition Workshops (CVPRW),2020 IEEE/CVF Conference on**, [s. l.], p. 3008–3017, 2020. DOI 10.1109/CVPRW50498.2020.00359. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.9150790&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 maio. 2021.

CUBUK, E. D. *et al.* AutoAugment: Learning Augmentation Strategies From Data. **2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision and Pattern Recognition (CVPR), 2019 IEEE/CVF Conference on**, [s. l.], p. 113–123, 2019. DOI 10.1109/CVPR.2019.00020. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.8953317&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 maio. 2021.

DMB, Demografia Médica no Brasil 2018 / Coordenação de Mário Scheffer; equipe de pesquisa: Alex Cassenote, Aline Gil Alves Guilloux, Aureliano Biancarelli, Bruno Alonso Miotto e Giulia Marcelino Mainardi. – São Paulo: Departamento de Medicina Preventiva da Faculdade de Medicina da USP; Conselho Regional de Medicina do Estado de São Paulo; Conselho Federal de Medicina, 2018.

DENG, L; YU, D. **Deep Learning: Methods and Applications**, Boston, Editora: Now Publishers,, 2014.

DUARTE, M.M. 2014. **Trade-off entre variância e viés e suas implicações na predição do estoque de carbono por florestas em processo de restauração.** In: Prado , P.I. & Batista, J.L.F. Modelagem Estatística para Ecologia e Recursos Naturais. Universidade de São Paulo. url: <http://cmq.esalq.usp.br/BIE5781>.

DUCHESNE, Pierre; RÉMILLARD, Bruno (Ed.). **Statistical modeling and analysis for complex data problems.** Springer Science & Business Media, 2005.

DUMOULIN, V.; VISIN, F. A guide to convolution arithmetic for deep learning. 2016.

DUCHESNE, Pierre; RÉMILLARD, Bruno (Ed.). **Statistical modeling and analysis for complex data problems.** Springer Science & Business Media, 2005

ESTEVA, A. (1) *et al.* Dermatologist-level classification of skin cancer with deep neural networks. **Nature**, [s. l.], v. 542, n. 7639, p. 115–118, [s. d.]. DOI 10.1038/nature21056. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-85016143105&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.

FERLAY, J. et al. (ed.). **Cancer today.** Lyon, France: International Agency for Research on Cancer, 2018. (IARC CANcerBase, n. 15). Disponível em: <https://publications.iarc.fr/Databases/Iarc-Cancerbases/Cancer-Today-Powered-By-GLOBOCAN-2018--2018>. Acessado em: 02 nov 2020.

FIREFLY. DE350 Wireless Polarizing Dermatoscope / Dermoscope. Disponível em: <https://fireflyglobal.com/de350-digital-dermatoscope/>. Acesso em: 01 abr. 2021.

FORSTER, M. 2000. **Key concepts in model selection:** Performance and generalizability. Journal of Mathematical Psychology, 44: 205–231. doi:10.1006/jmps.1999.1284

FEDERMAN, D. G. **Comparison of dermatologie diagnoses by primary care practitioners and dermatologists a review of the literature.** *Archives of Family Medicine*, v. 8, n. 2, p. 170–172, 1999.

FINLAYSON, G; TREZZI, E. "Shades of gray and colour constancy." 37-41. (2004).

GARG, S.; SINGH, P. State-of-the-Art Review of Deep Learning for Medical Image Analysis. **2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), Intelligent Sustainable Systems (ICISS), 2020 3rd International Conference on,** [s. l.], p. 421–427, 2020. DOI 10.1109/ICISS49785.2020.9315931. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.9315931&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow:** concepts, tools, and techniques to build intelligent systems. 2. ed. Canada: O'Reilly Media, 2019.

GESSEERT, N. *et al.* Skin Lesion Diagnosis using Ensembles, Unscaled Multi-Crop Evaluation and Loss Weighting. *[s. l.]*, 2018. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsair&AN=edsair.od.....18..a25d513bac83b10888c2f5759a1521c0&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. [S.l.]: Pearson, 2018. ISBN 9780133356724.

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

GUTMAN, D. *et al.* Skin Lesion Analysis toward Melanoma Detection: A Challenge at the International Symposium on Biomedical Imaging (ISBI) 2016, hosted by the International Skin Imaging Collaboration (ISIC). [s. l.], 2016. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.E80D1702&lang=pt-br&site=eds-live&scope=site>. Acesso em: 23 maio. 2021.

HARRISON, Matt. **Machine Learning: Guia de Referência Rápida**: trabalhando com dados estruturados em python. São Paulo: Novatec, 2020. 286 p. Tradução de: Lúcia A. Kinosita.

HASTIE, T. TIBSHIRANI, R. FRIEDMAN, J. 2009. **The elements of statistical learning**. Second edition. New York: Springer Series in Statistics Springer New York Inc. 739 p.

HE, K. *et al.* Deep Residual Learning for Image Recognition. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on**, [s. l.], p. 770–778, 2016. DOI 10.1109/CVPR.2016.90. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.7780459&lang=pt-br&site=eds-live&scope=site>. Acesso em: 11 mar. 2021.

HE, K. *et al.* Identity Mappings in Deep Residual Networks. [s. l.], 2016. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.E50DD2FD&lang=pt-br&site=eds-live&scope=site>. Acesso em: 11 mar. 2021.

HO, Y.; WOOKEY, S. The Real-World-Weight Cross-Entropy Loss Function: Modeling the Costs of Mislabeling. **IEEE Access, Access, IEEE**, [s. l.], v. 8, p. 4806–4813, 2020. DOI 10.1109/ACCESS.2019.2962617. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.8943952&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 fev. 2021.

HUANG, G. *et al.* Densely Connected Convolutional Networks. **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) ; ISBN 9781538604571**, [s. l.], 2017. DOI 10.1109/cvpr.2017.243. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.23C1239&lang=pt-br&site=eds-live&scope=site>. Acesso em: 11 mar. 2021.

HUBEL, D. H.; WIESEL, T. N. **Receptive fields, binocular interaction and functional architecture in the cat's visual cortex**. The Journal of Physiology, Wiley Online Library, v. 160, n. 1, p. 106–154, 1962.

HUBEL, D. H.; WIESEL, T. N. **Ferrier lecture-functional architecture of macaque monkey visual cortex**. Proceedings of the Royal Society of London. Series B. Biological Sciences, The Royal Society London, v. 198, n. 1130, p. 1–59, 1977.

- HAYKIN, Simon. **Redes Neurais: Princípios e prática.** Porto Alegre, RS: Bookman, 2001.
- INCA, INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. **Estimativa 2020: incidência de câncer no Brasil** / Instituto Nacional de Câncer José Alencar Gomes da Silva. – Rio de Janeiro: INCA, 2019. Disponível em: <https://www.inca.gov.br/sites/ufu.sti.inca.local/files/media/document/estimativa-2020-incidencia-de-cancer-no-brasil.pdf>
- INCA, INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. **Tipos de câncer.** Rio de Janeiro: INCA, 2019. Disponível em: <https://www.inca.gov.br/tipos-de-cancer>. Acesso em: 5 set. 2019.
- INCA, INSTITUTO NACIONAL DE CÂNCER JOSÉ ALENCAR GOMES DA SILVA. **ABC do câncer: abordagens básicas para o controle do câncer.** Rio de Janeiro: INCA, 2011. Disponível em: https://bvsms.saude.gov.br/bvs/publicacoes/abc_do_cancer.pdf. Acesso em: 08 mar. 2021.
- KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. [s. l.], 2014. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.F2AE122C&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 fev. 2021.
- KITCHENHAM, B.; CHARTERS, S. **Guidelines for performing systematic literature reviews in software engineering.** Keele University. Keele, p. 3-65. 2007.
- KRIES, J. von, “**Influence of adaptation on the effects produced by luminous stimuli,**” Sources of Color Vision, pp. 109–119, 1970.
- LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. **Gradient-based learning applied to document recognition.** *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998.
- LEE, Y. C.; JUNG, S.-H.; WON, H.-H. WonDerM: Skin Lesion Classification with Fine-tuned Neural Networks. [s. l.], 2018. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.7AAD9A03&lang=pt-br&site=eds-live&scope=site>. Acesso em: 23 maio. 2021.
- LIM, S. *et al.* Fast AutoAugment. [s. l.], 2019. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.8F08BFB6&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 maio. 2021.
- LITJENS, G. *et al.* A survey on deep learning in medical image analysis. **Medical Image Analysis**, [s. l.], v. 42, p. 60–88, 2017. DOI 10.1016/j.media.2017.07.005. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edselp&AN=S1361841517301135&lang=pt-br&site=eds-live&scope=site>. Acesso em: 1 abr. 2021.
- MAHBOD, A. *et al.* Investigating and Exploiting Image Resolution for Transfer Learning-based Skin Lesion Classification. [s. l.], 2020. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.6AA8556C&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 maio. 2021.

- MAHONY, N. O. *et al.* Deep Learning vs. Traditional Computer Vision. [s. l.], 2019. DOI 10.1007/978-3-030-17795-9. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsair&AN=edsair.od.....18..b2c155aec921b4e856df3acb3fed18f5&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.
- MARCELINO, Pedro. **Transfer learning from pre-trained models**. 2018. Disponível em: <https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751>. Acesso em: 10 out. 2021.
- MENZIES, S. W. *et al.* Short-term digital surface microscopic monitoring of atypical or changing melanocytic lesions. **Archives of dermatology**, [s. l.], v. 137, n. 12, p. 1583–1589, 2001. DOI 10.1001/archderm.137.12.1583. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=mdc&AN=11735708&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 abr. 2021.
- MENEZES, Nuno. Dermatoscopia De Lesões Pigmentadas. **Revista da Sociedade Portuguesa de Dermatologia e Venereologia**, [s. l.], v. 69, n. 1, 2016. DOI 10.29021/spdv.69.1.633. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsdoj&AN=edsdoj.4eec8cec8db4652a4c0df3786f75748&lang=pt-br&site=eds-live&scope=site>. Acesso em: 8 abr. 2021.
- MIAO, Hongbo. **How to plot precision and recall of multiclass classifier?** 2019. Disponível em: <https://stackoverflow.com/questions/56090541/how-to-plot-precision-and-recall-of-multiclass-classifier>. Acesso em: 23 jan. 2021.
- MUSTAIN, Muhamad. **Product Image Classification using Ensemble Learning**. 2020. Disponível em: <https://medium.com/@musmeong/product-image-classification-using-ensemble-learning-e5b6d166afc2>. Acesso em: 15 out. 2021.
- MOHAN, V. C.; DHARAN, S. A review on skin lesion classification techniques. *International journal of engineering research and technology*, v. 8, 2019.
- NEGRETE, Pablo David Minango. **Uso de técnicas de deep learning para classificação de imagens dermoscópicas com lesões na pele**. 2020. 1 recurso online (68 p.) Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP.
- NWANKPA, C. et al. **Activation Functions: Comparison of trends in Practice and Research for Deep Learning**. [s. l.], 2018. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.BC1F1089&lang=pt-br&site=eds-live&scope=site>. Acesso em: 5 jun. 2021.
- RAMPAZZO, Willian. Aprendizado profundo e interfaces cérebro-computador. 2019. 1 recurso online (99 p.). Dissertação (mestrado) - Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP.
- REZZE, G. G.; SÁ, B. C. S. de; NEVES, R. I. Dermatoscopia: o método de análise de padrões / Dermoscopy: the pattern analysis. **Anais Brasileiros de Dermatologia**, [s. l.], v. 81, n. 3, p. 261–268, 2006. DOI 10.1590/S0365-05962006000300009. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edssci&AN=edssci.S0365.05962006000300009&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 abr. 2021.

REZZE, Gisele Garganti; PASCHOAL, Francisco Macedo; HIRATA, Sergio Henrique. **Atlas De Dermatoscopia Aplicada.** 2. ed. São Paulo: Lemar, 2014. 192 p. ISBN: 9788586652417.

ROCHA, Carlos Vinicios Martins; MAIA, Lucas Bezerra; JUNIOR, Geraldo Braz; DE ALMEIDA, João Dallyson Sousa; DE PAIVA, Anselmo Cardoso. **Otimização de CNN aplicada ao Diagnóstico de Melanoma.** In: ANAIS PRINCIPAIS DO SIMPÓSIO BRASILEIRO DE COMPUTAÇÃO APLICADA À SAÚDE (SBCAS), 19., 2019, Niterói. *Anais* [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 336-341. DOI: <https://doi.org/10.5753/sbcas.2019.6272>. Acesso em: 31 out. 2020.

RUSSAKOVSKY, O. et al. **Imagenet large scale visual recognition challenge.** International Journal of Computer Vision, [S.I.], v.115, n.3, p.211–252, 2015

SAE-LIM, W.; WETTAYAPRASIT, W.; AIYARAK, P. Convolutional Neural Networks Using MobileNet for Skin Lesion Classification. **2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE), Computer Science and Software Engineering (JCSSE), 2019 16th International Joint Conference on,** [s. l.], p. 242–247, 2019. DOI 10.1109/JCSSE.2019.8864155. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.8864155&lang=pt-br&site=eds-live&scope=site>. Acesso em: 3 jun. 2021.

SANTOS, André Pires dos. **Análise e comparação do desempenho de diferentes arquiteturas de redes neurais artificiais profundas aplicadas a triagem de lesões de pele.** 2019. 95 f. Dissertação (Mestrado em Engenharia Elétrica e Computação) - Universidade Presbiteriana Mackenzie, São Paulo, 2019.

SADDLE point. 2016. Elaborado por: Nicoguaro. Disponível em: https://en.wikipedia.org/wiki/Saddle_point. Acesso em: 11 maio 2021.

SBCD, Sociedade Brasileira de Cirurgia Dermatológica. Acesso em novembro de 2020, disponível em <https://www.sbcd.org.br/cirurgia-dermatologica/o-que-e-cirurgia-dermatologica/para-sua-pele/cancer-de-pele/>.

SCHEFFER, M.; CASSENTE, A.; GUILLOUX, A. G. A.; MIOTTO, B. A.; MAINARDI, G. M. **Demografia Médica no Brasil.** Brasilia São Paulo, SP: FMUSP, Cremesp, p. 1-35. 2018.

SCHMITT, J. V.; MIOT, H. A.; SCHMITT, J. V.; MIOT, H. A. **Distribution of Brazilian dermatologists according to geographic location, population and HDI of municipalities: an ecological study.** *Anais Brasileiros de Dermatologia.* Rio de Janeiro. v. 89, n. 6, p. 1013–1015, dez. 2014.

SCTIE, Ministério da Saúde: Secretaria de Ciência, Tecnologia e Insumos Estratégicos. Departamento de Ciência e Tecnologia. **Diretrizes metodológicas: Elaboração de revisão sistemática e metanálise de estudos de acurácia diagnóstica.** Ministério da Saúde, Secretaria de Ciência, Tecnologia e Insumos Estratégicos, Departamento de Ciência e Tecnologia. – Brasília: Editora do Ministério da Saúde, 2014.116 p. : il. ISBN 978-85-334-2129-5

SCUDILIO, Juliana. **Qual a melhor métrica para avaliar os modelos de Machine Learning?** 2020. Disponível em: <https://www.flai.com.br/juscudilio/qual-a-melhor-metrica-para-avaliar-os-modelos-de-machine-learning/>. Acesso em: 18 jan. 2021.

SCHADE, Gabriel. **Machine Learning**: métricas para modelos de classificação. Métricas para Modelos de Classificação. 2019. Disponível em: <https://gabriel-schade.github.io/2019/03/12/ml-classificacao-metricas.html>. Acesso em: 12 jan. 2021.

SERTE, S.; SERENER, A.; ALÂTURJMAN, F. Deep learning in medical imaging: A brief review. **Transactions on Emerging Telecommunications Technologies ; ISSN 2161-3915 2161-3915**, [s. l.], 2020. DOI 10.1002/ett.4080. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.4AA1AE5F&lang=pt-br&site=eds-live&scope=site>. Acesso em: 1 abr. 2021.

SHEN, S *et al.* Low-cost and high-performance data augmentation for deep-learning-based skin lesion classification. [s. l.], 2021. Disponível em: <https://arxiv.org/abs/2101.02353>. Acesso em: 23 maio. 2021.

SMITH, L. N. **Cyclical Learning Rates for Training Neural Networks**. 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on, WACV, [s. l.], p. 464–472, 2017. DOI 10.1109/WACV.2017.58. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.7926641&lang=pt-br&site=eds-live&scope=site>. Acesso em: 5 jun.

SOUTOR, Carol; HORDINKY, Maria K.. **Dermatologia clínica**. Porto Alegre: Amgh, 2015. 364 p. Tradução: Ademar Valadares Fonseca; Revisão técnica: Tania Ludmila de Assis. ISBN 978-85-8055-380-2.

SZEGEDY, C. *et al.* Rethinking the Inception Architecture for Computer Vision. [s. l.], 2015. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsair&AN=edsair.de-dup.wf.001..e3ccf9935e81d95ffbc9afc5f36a2092&lang=pt-br&site=eds-live&scope=site>. Acesso em: 11 mar. 2021.

SZEGEDY, C. *et al.* Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. [s. l.], 2016. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.67F8A59D&lang=pt-br&site=eds-live&scope=site>. Acesso em: 14 abr. 2021.

TAN, M.; LE, Q. V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. [s. l.], 2019. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.C5EA3A6E&lang=pt-br&site=eds-live&scope=site>. Acesso em: 14 abr. 2021.

TSANG, Sik-Ho. **Review: Xception**: with depthwise separable convolution, better than inception-v3 (image classification). With Depthwise Separable Convolution, Better Than Inception-v3 (Image Classification). 2018. Disponível em: <https://towardsdatascience.com/review-xception-with-depthwise-separable-convolution-better-than-inception-v3-image-dc967dd42568>. Acesso em: 11 mar. 2021.

TSCHANDL P., ROENDAHL C. , KITTLER H. **The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions**. *Sci. Data* **5**, 180161 doi.10.1038/sdata.2018.161 (2018).

- UPADHYAY, Amit. **Precision/Recall Tradeoff**. 2020. Disponível em: <https://medium.com/analytics-vidhya/precision-recall-tradeoff-79e892d43134>. Acesso em: 20 jan. 2021.
- VASCONCELLOS, Paulo. **Como saber se seu modelo de Machine Learning está funcionando mesmo**. 2018. Disponível em: <https://paulovasconcellos.com.br/como-saber-se-seu-modelo-de-machine-learning-est%C3%A1-funcionando-mesmo-a5892f6468b>. Acesso em: 09 jan. 2021.
- VESTERGAARD, M. E. (1) *et al.* Dermoscopy compared with naked eye examination for the diagnosis of primary melanoma: A meta-analysis of studies performed in a clinical setting. **British Journal of Dermatology**, [s. l.], v. 159, n. 3, p. 669–676, [s. d.]. DOI 10.1111/j.1365-2133.2008.08713.x. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edselc&AN=edselc.2-52.0-49749089750&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.
- VIEIRA, E. W. R.; LIMA, T. M. N.; GAZZINELLI, A. **The Waiting Time for Specialized Medical Consultations in a Small Municipality of Minas Gerais, Brazil**. **REME: Revista Mineira de Enfermagem**, v. 19, n. 1, p. 65–71, 2015.
- VON ZUBEN, F. J.; VERONEZE, R. **Deep Learning: parte I**. 2019. Notas de aula do curso de pós-graduação IA353 - Redes Neurais, Primeiro Semestre, FEEC, Unicamp.
- VUPPALA, S. K. *et al.* Explainable Deep Learning Methods for Medical Imaging Applications. **2020 IEEE 5th International Conference on Computing Communication and Automation (ICCCA), Computing Communication and Automation (ICCCA), 2020 IEEE 5th International Conference on**, [s. l.], p. 334–339, 2020. DOI 10.1109/ICCCA49541.2020.9250820. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.9250820&lang=pt-br&site=eds-live&scope=site>. Acesso em: 11 abr. 2021.
- WANG, Y.; WEI, G.; BROOKS, D. *Benchmarking tpu, gpu, and CPU platforms for deep learning*. CoRR, abs/1907.10701, 2019. Disponível em: <<http://arxiv.org/abs/1907.10701>>.
- WOLFF, Klaus. **Dermatologia de Fitzpatrick**: atlas e texto. 8. ed. Porto Alegre: Amgh, 2019. 898 p. Tradução: André Garcia Islabão; revisão técnica: Gabriela Fortes Escobar, Renan Rangel Bonamigo.
- YAO, P. *et al.* Single Model Deep Learning on Imbalanced Small Datasets for Skin Lesion Classification. [s. l.], 2021. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.232CF58A&lang=pt-br&site=eds-live&scope=site>. Acesso em: 2 maio. 2021.
- YOSINSKI, J. *et al.* How transferable are features in deep neural networks? [s. l.], 2014. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.1466255C&lang=pt-br&site=eds-live&scope=site>. Acesso em: 13 abr. 2021.
- YU, L. *et al.* Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks. **IEEE Transactions on Medical Imaging, Medical Imaging, IEEE Transactions on, IEEE Trans. Med. Imaging**, [s. l.], v. 36, n. 4, p. 994–1004, 2017. DOI

10.1109/TMI.2016.2642839. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.7792699&lang=pt-br&site=eds-live&scope=site>. Acesso em: 4 jul. 2021.

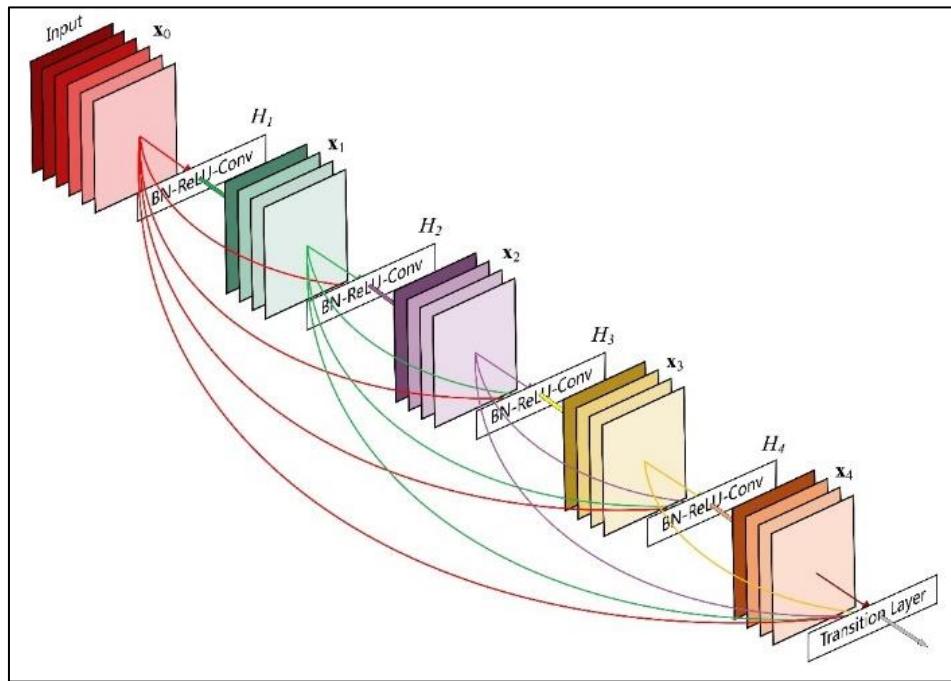
ZHOU, B. *et al.* Learning Deep Features for Scene Recognition using Places Database. **MIT web domain**, [s. l.], 2014. Disponível em: <https://search.ebscohost.com/login.aspx?direct=true&db=edsbas&AN=edsbas.EB31A0D0&lang=pt-br&site=eds-live&scope=site>. Acesso em: 13 abr. 2021.

APÊNDICE A. DETALHAMENTO DAS ARQUITETURAS DE REDES NEURAIS CONVOLUCIONAIS PROFUNDAS

A.1. DENSENET

A arquitetura DenseNet (HUANG et al., 2017) utiliza uma conectividade de camada cruzada, mas de uma forma modificada. A conexão ocorre a cada camada precedente para a próxima camada de uma forma *feed-forward*. Assim, os mapas de características de todas as camadas anteriores são utilizados como entradas em todas as camadas subsequentes (Figura 39). Um bloco denso com 5 camadas e taxa de crescimento 4). Dessa maneira, garantindo o máximo fluxo de informação entre as camadas da rede, conectados todas as camadas (com tamanhos de mapa de recursos correspondentes) diretamente entre si.

Figura 39. Um bloco denso com 5 camadas e taxa de crescimento 4



Fonte: Huang et al. (2017).

Por conseguinte, preservando a natureza do *feed-forward*, cada camada obtém entradas adicionais de todas as camadas anteriores e passa em seus próprios mapas de características para as camadas. Ainda mais, uma importante variável dentro dos blocos densos é a taxa de crescimento, que regula quanta informação nova, cada camada irá contribuir para o estado global. Bem como, com as arquiteturas tradicionais, as CNNs têm camadas L com conexões L,

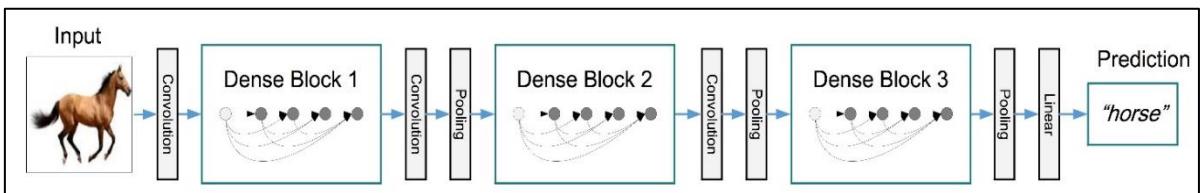
uma entre cada camada e suas subsequentes camadas, já a DenseNet tem $L(L + 1)/2$ conexões diretas entre cada camada (HUANG et al., 2017).

Algumas vantagens da DenseNet que tornam essa rede atraente são: Lidar com o problema do gradiente de desaparecimento, fortalece a propagação de mapas de recursos, incentiva a reutilização de mapas de recursos e reduz substancialmente o número de parâmetros (HUANG et al., 2017).

Esse problema de gradiente de desaparecimento acontece quando uma rede se torna muito profunda, ou seja, tem muitas camadas, quando se propaga o erro de volta para a rede, esse erro é reduzido a cada etapa e eventualmente torna-se 0. Assim sendo, os gradientes de onde a função de perda são calculados diminuem facilmente para zero após várias aplicações da regra da cadeia. Isso resulta que, os pesos não atualizam seus valores e, portanto, nenhum aprendizado está sendo realizado (GOODFELLOW; BENGIO e COURVILLE. 2016).

Portanto, a DenseNet, resolve essa questão, incentivando a reutilização de recursos e reduzindo assim, o número de parâmetros, pelo mesmo motivo, uma vez que estão reutilizando informações de mapas de recursos anteriores em vez de gerar mais parâmetros. Na Figura 40, é possível ver a esquemática de uma DenseNet com três blocos densos. As camadas entre dois blocos densos adjacentes são chamadas de camadas de transição, elas têm como função, a mudança dos tamanhos de mapas de características via convolução e *pooling*.

Figura 40. Uma DenseNet profunda com três blocos densos.



Fonte: HUANG et al. (2017).

No Quadro 13, é demonstrado algumas das formas de arquiteturas da DenseNet, os valores representados como tamanho de saída se dão devido ao tamanho da imagem de entrada serem do ImageNet. Todas as redes têm uma taxa de crescimento (um *hyperparameter* da arquitetura) de 32, responsável pelo número de conexões com as camadas L+1. Vale ressaltar que para cada camada "conv" mostrada no Quadro 3 correspondem a uma sequência BN-ReLU-

Conv, ou seja, camada batchNormalization, camada ReLU e Convolução. O tamanho dos blocos densos é a única coisa que muda para cada versão da arquitetura da DenseNet.

Quadro 13. Arquiteturas DenseNet para ImageNet. A taxa de crescimento para todas as redes é $k = 32$. Observe que cada camada "conv" mostrada abaixo corresponde a sequência BN-ReLU-Conv

Camadas	Tamanho da saída	DenseNet-121	DenseNet-169	DenseNet-201
Convolução	112×112		7×7 conv, stride 2	
<i>Pooling</i>	56×56		3×3 max pool, stride 2	
Bloco Denso (1)	56×56	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 6$
Bloco de transição (1)	56×56		1×1 conv	
	28×28		3×3 average pool, stride 2	
Bloco Denso (2)	28×28	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 12$
Bloco de transição (2)	28×28		1×1 conv	
	14×14		3×3 average pool, stride 2	
Bloco Denso (3)	14×14	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 48$
Bloco de transição (3)	14×14		1×1 conv	
	7×7		3×3 average pool, stride 2	
Bloco Denso (4)	7×7	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 & conv \\ 3 \times 3 & conv \end{bmatrix} \times 32$
Camada de classificação	1×1		7×7 global average pool	
			1000D totalmente-conectada, softmax	

Fonte: Adaptado de HUANG et al. (2017).

A.2. RESNET-V2

Como mencionado na sessão anterior, um grande problema ao se aumentar a profundidade de uma rede é que se torna difícil o treinamento dela. Além disso, notou-se que depois de alguma profundidade, o desempenho se degrada. Levando ao gradiente de desaparecimento. Como resultado, conforme a rede se aprofunda, seu desempenho fica saturado. Em virtude disso, a arquitetura ResNet-V2 (HE et al., 2016), apresenta uma estrutura de aprendizado residual que facilita o treinamento de redes que são substancialmente mais profundas. O resíduo pode ser simplesmente entendido como a subtração do recurso aprendido da entrada dessa camada.

Como as redes neurais são bons aproximadores de funções, elas devem ser capazes de resolver facilmente a função de identificação, em que a saída de uma função se torna a própria entrada, demonstrado na Equação 16:

$$f(x) = x \quad (16)$$

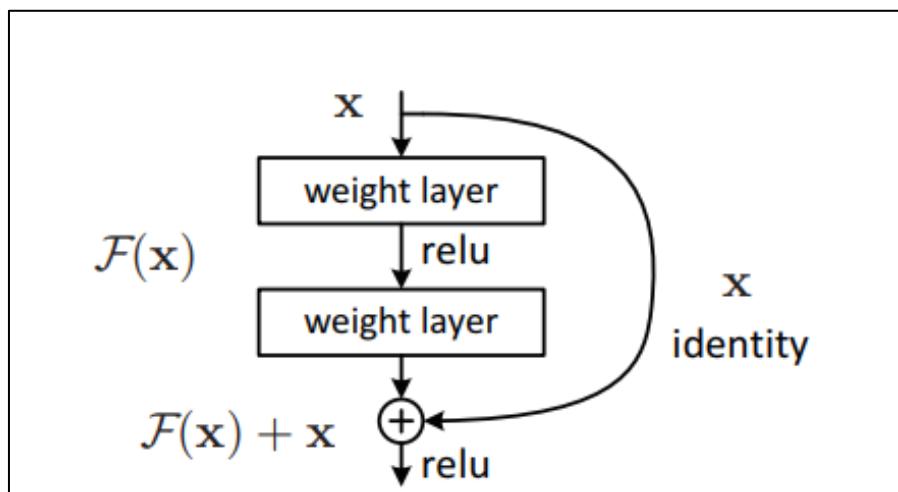
Seguindo a mesma lógica, contornando a entrada para a primeira camada do modelo para ser a saída da última camada do modelo, a rede deve ser capaz de prever qualquer função que estava aprendendo antes com a entrada adicionada a ela (Equação 17):

$$f(x) + x = h(x) \quad (17)$$

A intuição é que aprender $f(x) = 0$ deve ser fácil para a rede (HE et al., 2016).

Sobretudo a ResNet, possui dois tipos de blocos residuais, sendo o primeiro, exposto na Figura 41. Esse é o mais simples entre os dois, pois conta com duas ou mais convoluções cuja saída final possui tamanho compatível com a entrada, após passar pelas camadas do bloco o resultado é simplesmente somado, elemento a elemento e canal a canal à entrada do bloco.

Figura 41. Aprendizagem residual: um bloco de construção



Fonte: HE et al., (2016).

Isto é, o bloco presente na Figura 41 é composto por duas camadas convolucionais, para a primeira camada manter a saída com as mesmas dimensões, altura e largura, da entrada utiliza-se um método chamado *padding*. Todavia, ao longo de CNN, espera-se que a entrada tenha seu tamanho reduzido, ao passo que os números de camadas aumentem.

Esse comportamento é encontrado no segundo tipo de bloco residual, uma pequena mudança no bloco da Figura 41, adicionando uma convolução no caminho da identidade que reduza seu tamanho a dimensões apropriadas e aumente o número de canais, tornando a compatível para a soma com $F(x)$.

Por fim, no Quadro 14 são demonstradas algumas das formas de arquiteturas da ResNet-V1, os valores representados como tamanho de saída se dão devido ao tamanho da imagem de entrada da rede ser 224 x 224 devido aos dados de treinamentos serem da ImageNet.

Quadro 14. Arquiteturas ResNet para ImageNet. Os blocos de construção são mostrados entre parênteses, com os números de blocos empilhados. A redução da resolução é realizada por conv3 1, conv4 1 e conv5 1 com uma passada de 2.

Camadas	Tamanho da saída	ResNet-50	ResNet-101	ResNet-152
Conv1	112×112	$7 \times 7, 64, \text{stride } 2$		
Conv2_x	56×56	$3 \times 3 \text{ max pool, stride } 2$		
		$\begin{bmatrix} 1 \times 1 & ,64 \\ 3 \times 3 & ,64 \\ 1 \times 1 & ,256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & ,64 \\ 3 \times 3 & ,64 \\ 1 \times 1 & ,256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & ,64 \\ 3 \times 3 & ,64 \\ 1 \times 1 & ,256 \end{bmatrix} \times 3$
Conv3_x	28×28	$\begin{bmatrix} 1 \times 1 & ,128 \\ 3 \times 3 & ,128 \\ 1 \times 1 & ,512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & ,128 \\ 3 \times 3 & ,128 \\ 1 \times 1 & ,512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1 & ,128 \\ 3 \times 3 & ,128 \\ 1 \times 1 & ,512 \end{bmatrix} \times 8$
Conv4_x	14×14	$\begin{bmatrix} 1 \times 1 & ,256 \\ 3 \times 3 & ,256 \\ 1 \times 1 & ,1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 & ,256 \\ 3 \times 3 & ,256 \\ 1 \times 1 & ,1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1 & ,256 \\ 3 \times 3 & ,256 \\ 1 \times 1 & ,1024 \end{bmatrix} \times 36$
Conv5_x	7×7	$\begin{bmatrix} 1 \times 1 & ,512 \\ 3 \times 3 & ,512 \\ 1 \times 1 & ,2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & ,512 \\ 3 \times 3 & ,512 \\ 1 \times 1 & ,2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1 & ,512 \\ 3 \times 3 & ,512 \\ 1 \times 1 & ,2048 \end{bmatrix} \times 3$
Camada de classificação	1×1	$7 \times 7 \text{ global average pool}$		
		1000D totalmente-conectada, softmax		

Fonte: Adaptado de He et al., (2016).

Observando a arquitetura da rede, nada mais é, do que a combinação dos dois blocos básicos seguidos, alternando de acordo com a presença ou não de *downsample* da entrada, ao final temos uma camada de *global average pool* e a camada de classificação.

A.3. XCEPTION

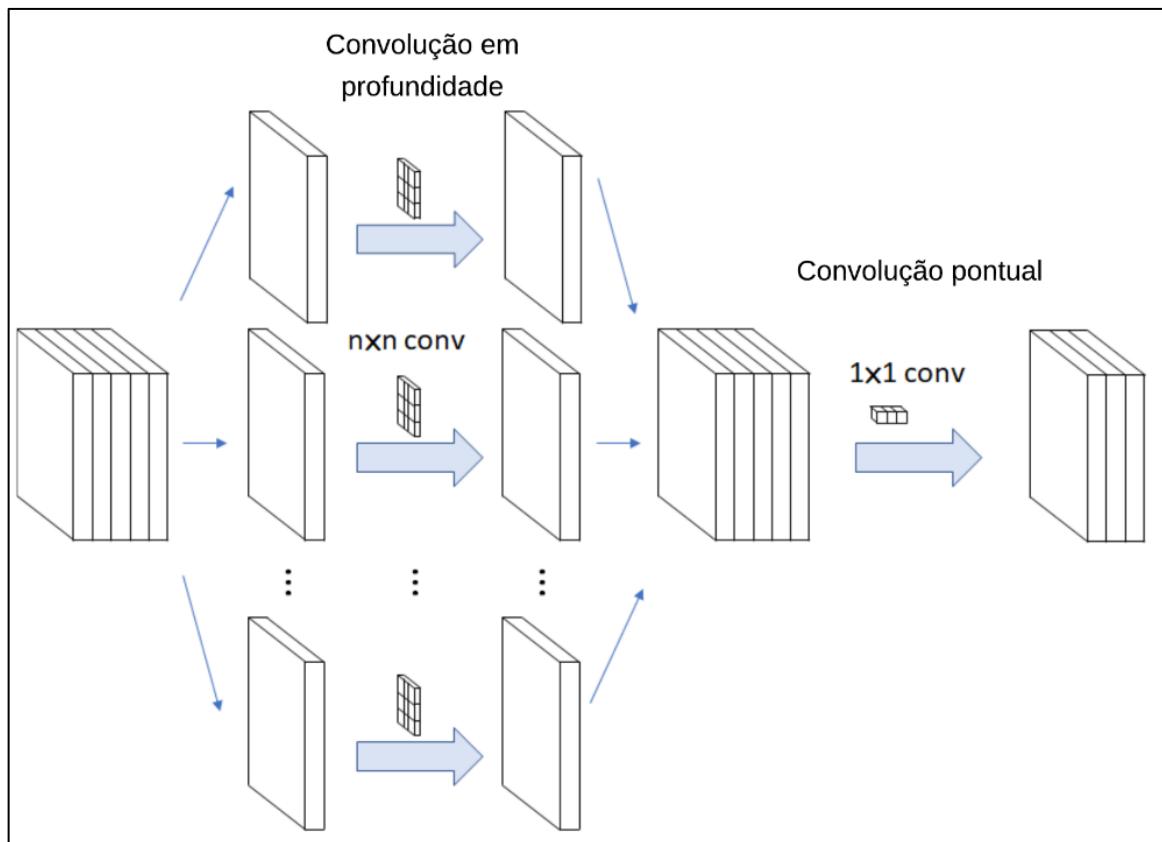
A arquitetura Xception, proposta por Chollet (2017), significa “Extreme Inception”. Para entender sua arquitetura, primeiramente é preciso entender o que é convolução separável em profundidade. A convolução separável em profundidade original, como apresentado na Figura 42, é a convolução em profundidade seguida por uma convolução pontual (TSANG, 2018).

- A convolução em profundidade é a convolução espacial $n \times n$ em nível de canal. Suponha que na Figura 14, existam 5 canais, então irá ter $n \times n \times 5$ convolução espacial. A convolução em profundidade, em vez de aplicar a convolução de tamanho $d \times d \times C$, será aplicada uma convolução de tamanho $d \times d \times 1$. Em outras palavras, não se utiliza o cálculo da convolução em todos os canais, mas apenas 1 por 1.

- A convolução pontual, na verdade, é a convolução 1×1 para alterar a dimensão.

Em comparação com a convolução convencional, não é necessário realizar a convolução em todos os canais. Isso significa que o número de conexões é menor e o modelo é mais leve (TSANG, 2018).

Figura 42. Convolução Separável em Profundidade Original



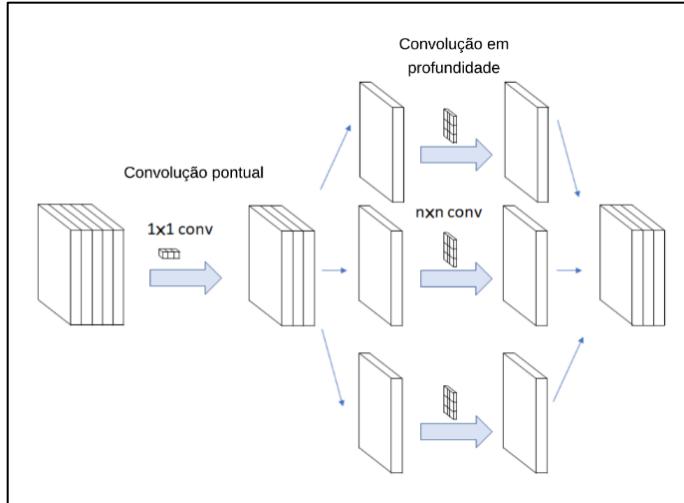
Fonte: Adaptado de TSANG (2018).

A convolução separável em profundidade modificada (Figura 43) é a convolução pontual seguida por uma convolução em profundidade. Esta modificação é motivada pelo módulo de iniciação no Inception-v3 que a convolução 1×1 é feita primeiro antes de quaisquer $n \times n$ convoluções espaciais. Dessa maneira, duas pequenas diferenças surgem (TSANG, 2018):

- A ordem das operações: Afirma-se que isso não é importante porque, quando é usado em uma configuração empilhada, há apenas pequenas diferenças que aparecem no início e no final de todos os módulos iniciais encadeados.

- A Presença / Ausência de Não Linearidade: No Módulo de Iniciação original, não há linearidade após a primeira operação. Em Xception, a convolução separável em profundidade modificada, não há nenhuma não linearidade ReLU intermediária.

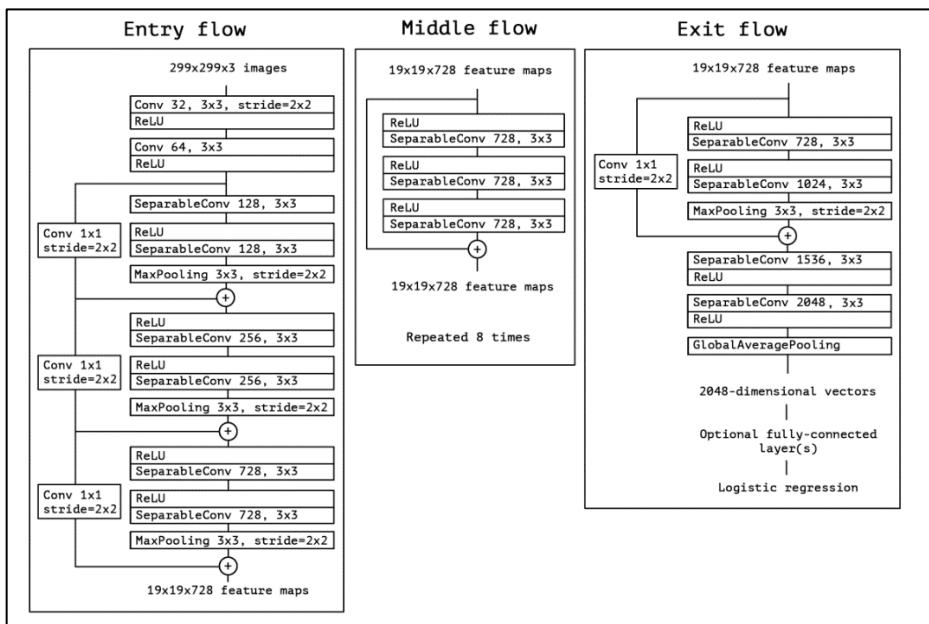
Figura 43. A convolução separável em profundidade modificada



Fonte: Adaptado de Tsang (2018).

A arquitetura da Xception é demonstrada na Figura 44. Os dados passam primeiro pelo fluxo de entrada, depois pelo fluxo intermediário, que é repetido oito vezes, e, finalmente, através do fluxo de saída. Observando que todas as camadas de Convolução e Convolução Separável (*SeparableConv*) são seguidas pela *batch Normalization* (não incluído no diagrama).

Figura 44. A arquitetura da Xception para o ImageNet challenge



Fonte: Chollet (2017).

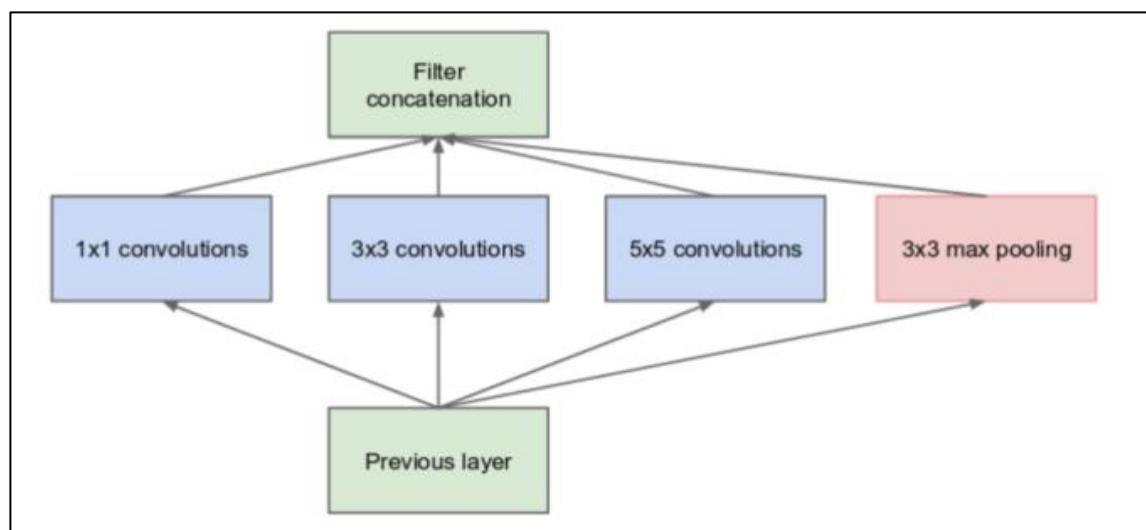
Todas as camadas Convolução Separável usam um multiplicador de profundidade de 1 (sem expansão de profundidade) (TSANG, 2018). A Convolução Separável (SeparableConv) é a convolução separável em profundidade modificada. Pode-se ver que SeparableConvs são tratados como módulos de iniciação e colocados em toda a arquitetura de aprendizado profundo (CHOLLET, F., 2017).

A.4. INCEPTIONV3

Em geral, cada camada de uma CNN extrai algum recurso, empilhar essas camadas uma sobre a outra não é uma boa ideia. Redes profundas são propensas a *overfitting*, e o encadeamento de várias operações convolucionais aumenta o custo de treinamento da rede, podendo levar ao desaparecimento de gradiente, como comentado nas seções anteriores. Outro problema é que, à medida que cada tipo de camada extrai um tipo diferente de informação, como saber quais filtros fornecem as informações mais úteis para a rede?

A arquitetura InceptionV3 (SZEGEDY et al., 2015), apresenta uma resposta para essa questão. Um módulo Inception calcula vários filtros diferentes, como apresentado na Figura 45, sobre a mesma entrada e, em seguida, concatena todas as saídas, o que permite ao modelo decidir quais recursos usar. Dessa forma, a Inception apresenta filtros com vários tamanhos operando no mesmo nível, tornando a arquitetura mais larga em vez de mais profunda (SZE-GEDY et al., 2015).

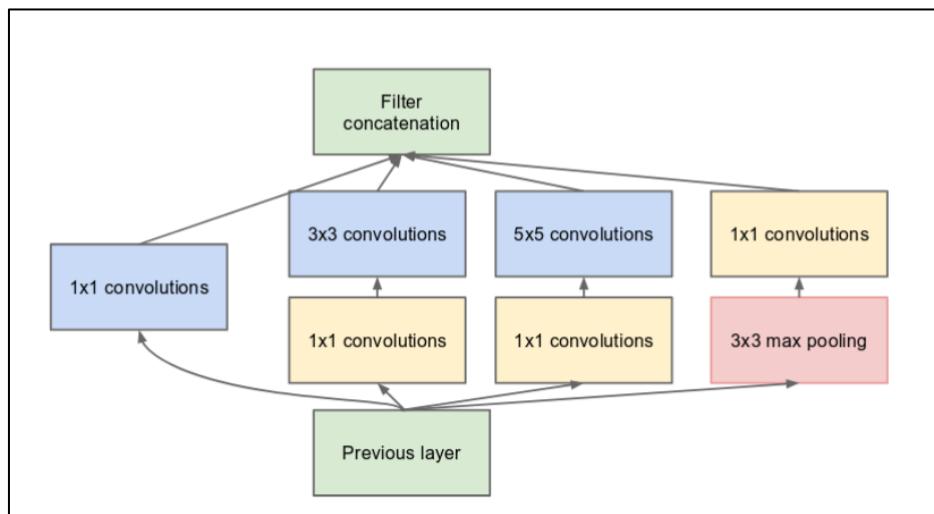
Figura 45. Módulo Inception Simples



Fonte: SZEGEDY et al. (2015).

Como afirmado antes, as redes neurais profundas são computacionalmente caras. Para torná-las mais baratas, os autores da Inception limitam o número de canais de entrada adicionando uma convolução 1×1 extra antes das convoluções 3×3 e 5×5 . Embora adicionar uma operação extra possa parecer contraintuitivo, as convoluções 1×1 são muito mais baratas do que as convoluções 5×5 , e o número reduzido de canais de entrada também ajuda. Observe que, no entanto, a convolução 1×1 é introduzida após a camada de *pooling* máximo, e não antes (Figura 46). Por fim, sua arquitetura é apresentada no Quadro 15, e a composição de cada módulo presente na arquitetura é exposta na Figura 47.

Figura 46. Módulo Inception com redução de dimensões



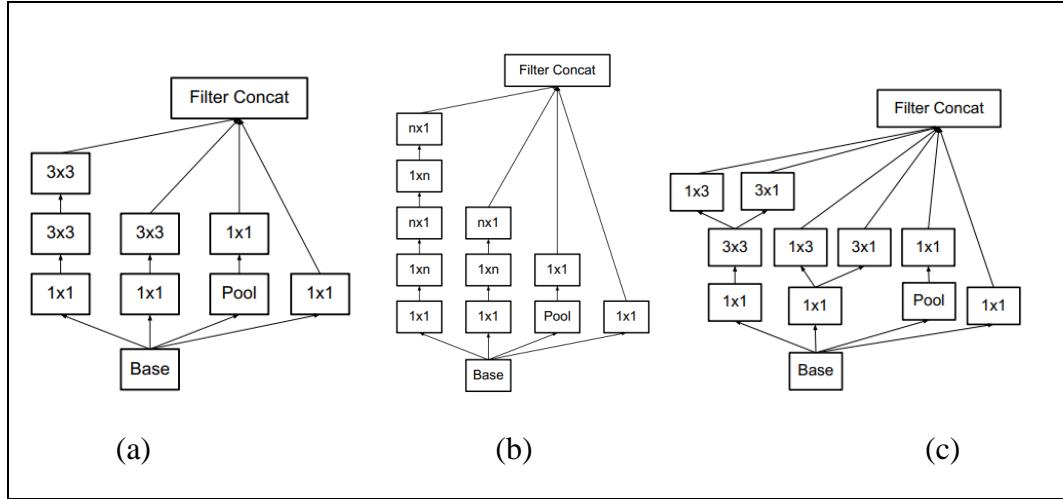
Fonte: Szegedy et al. (2015).

Quadro 15. A arquitetura InceptionV2/V3 proposta para o ImageNet challenge. O tamanho de saída de cada módulo é o tamanho de entrada do próximo.

Tipo de camada	Dimensões dos filtros	Stride	Tamanho da entrada
Conv	3×3	2	$299 \times 299 \times 3$
Conv	3×3	1	$149 \times 149 \times 32$
Conv com padded	3×3	1	$147 \times 147 \times 43$
Pooling	3×3	2	$147 \times 147 \times 64$
Conv	3×3	1	$73 \times 73 \times 64$
Conv	3×3	2	$71 \times 71 \times 80$
Conv	3×3	1	$35 \times 35 \times 192$
3 x Módulo Inception	Figura 27 - a		$35 \times 35 \times 288$
5x Módulo Inception	Figura 27 - b		$17 \times 17 \times 768$
2 x Módulo Inception	Figura 27 - c		$8 \times 8 \times 1280$
Pooling	8×8		$8 \times 8 \times 2048$
Linear	Logits		$1 \times 1 \times 2048$
Softmax	Classifier		$1 \times 1 \times 1000$

Fonte: Adaptado de Szegedy et al. (2015).

Figura 47. Tipos de módulos Inception



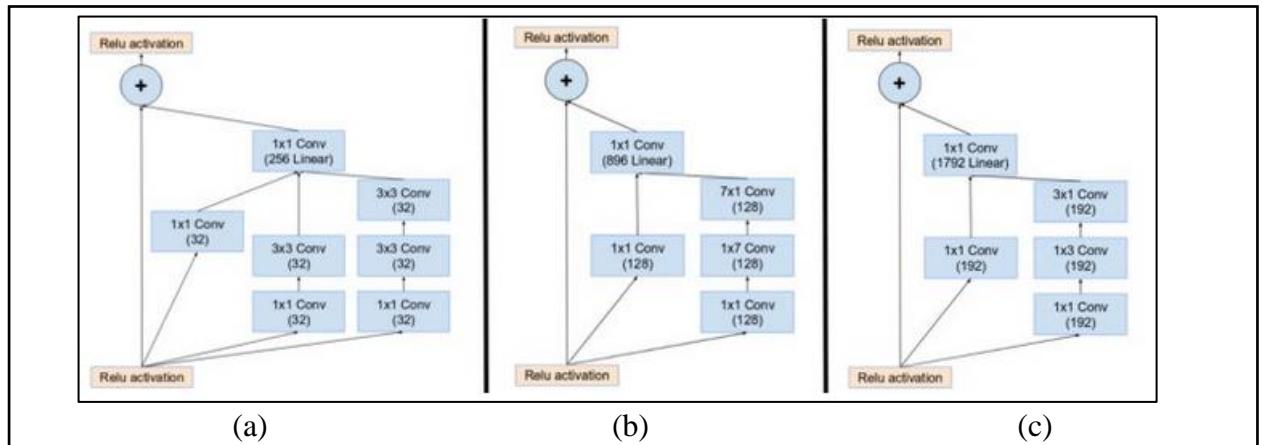
Fonte: Adaptado de Szegedy et al. (2015).

A.5. INCEPTIONRESNETV2

Inspirado no desempenho da ResNet, um modelo híbrido com Inception foi proposto por Szegedy et al. (2016). Essa arquitetura tem como objetivo introduzir as conexões residuais que adicionam a saída da operação de convolução no módulo Inception de entrada.

Para que a adição residual funcione, a entrada e a saída após a convolução devem ter as mesmas dimensões. Portanto, serão utilizadas convoluções 1x1 após as convoluções originais, para corresponder aos tamanhos de profundidade (a profundidade é aumentada após a convolução), como demonstrado na Figura 48 (SZEGEDY et al., 2016).

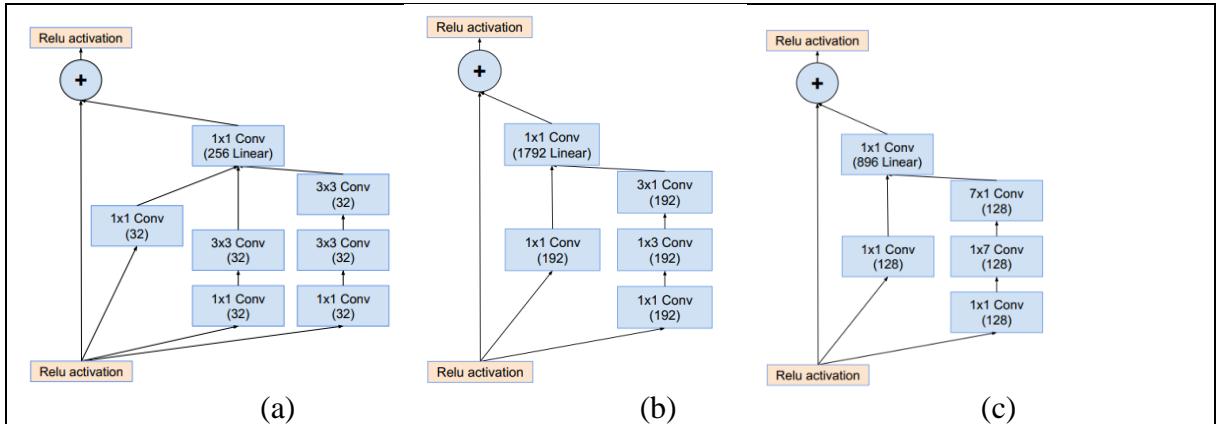
Figura 48. Módulos de iniciação A, B, C em um Inception-ResNet-V2 Observe como a camada de *pooling* foi substituída pela conexão residual e a convolução 1x1 adicional antes da adição.



Fonte: SZEGEDY et al., (2016).

A operação de *pooling* dentro dos módulos iniciais principais foi substituída em favor das conexões residuais (Figura 49). No entanto, ainda pode-se encontrar essas operações nos blocos de redução (SZEGEDY et al., 2016).

Figura 49. Bloco Residual Inception A (redução de tamanho de 35×35 a 17×17), Bloco Residual Inception B (redução de tamanho de 17×17 a 8×8) e Bloco Residual Inception C (tamanho de 8×8).

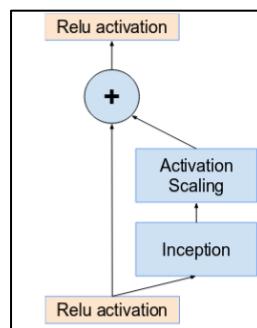


Fonte: Szegedy et al., (2016).

Um ponto que vale ser mencionado, é que as redes com unidades residuais mais profundas na arquitetura faziam com que a rede “morresse” se o número de filtros excedesse 1000. As variantes residuais começam a exibir instabilidades e a rede apenas “morre” no início do treinamento, o que significa que a última camada antes do agrupamento médio começou a produzir apenas zeros após algumas dezenas de milhares de iterações.

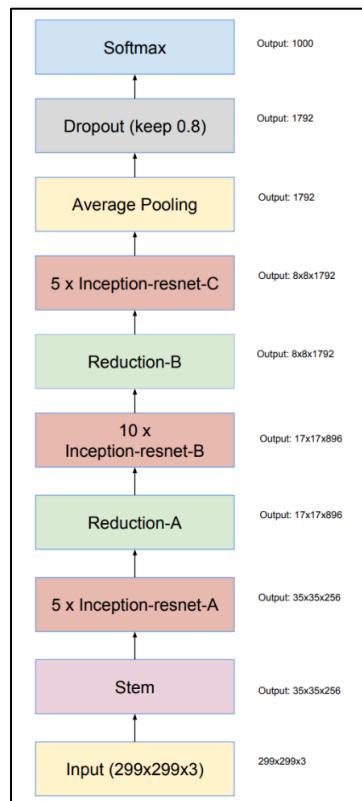
Isso não pôde ser evitado, nem reduzindo a taxa de aprendizado, nem adicionando uma *Batch normalization* extra a essa camada. Portanto, para aumentar a estabilidade, os autores escalam as ativações residuais em um valor em torno de 0,1 a 0,3, como demonstrado na Figura 50. Por fim, sua arquitetura completa é demonstrada na Figura 51 (SZEGEDY et al., 2016).

Figura 50. As ativações são escaladas por uma constante para evitar que a rede morra.



Fonte: Szegedy et al., (2016).

Figura 51. A arquitetura da Inception-ResNet para o ImageNet challenge.



Fonte: Szegedy et al., (2016).

A.6. EFFICIENTNET

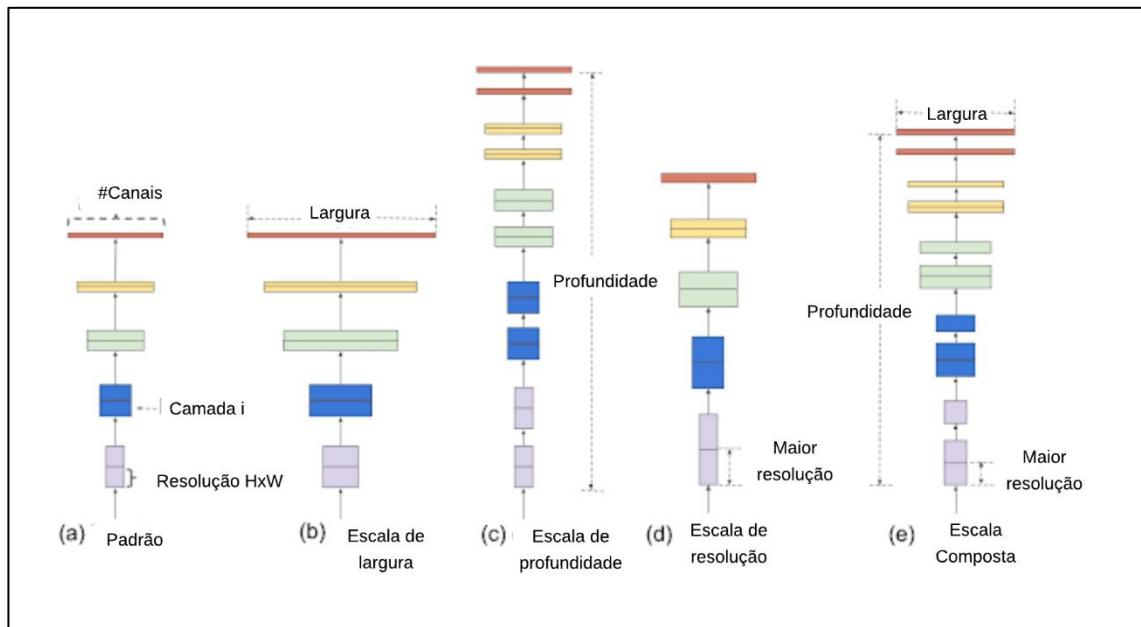
A arquitetura EfficientNet (TAN; LE, 2019), busca identificar quais os valores que equilibram cuidadosamente a profundidade, largura e resolução da rede, ocasionando um melhor desempenho.

Assim, foi proposto um método de dimensionamento composto para dimensionar as dimensões da rede. A estratégia *grid search* é aplicada para encontrar o relacionamento entre as diferentes dimensões de escala da rede de linha de base sob uma restrição de recursos fixa. Usando essa estratégia, é possível encontrar os coeficientes de escala apropriados para cada uma das dimensões a serem aumentadas. Usando esses coeficientes, a rede de linha de base foi dimensionada de acordo com o tamanho desejado.

Existem três dimensões de escala de uma CNN: profundidade, largura e resolução. Profundidade significa simplesmente a profundidade das redes, o que equivale ao número de camadas dela. Largura significa simplesmente a largura da rede. Uma medida de largura, por exemplo, é o número de canais em uma camada Conv, enquanto a Resolução é simplesmente a

resolução da imagem que está sendo passada para uma CNN. Na Figura 52, é demonstrado um comparativo entre os diferentes métodos de dimensionamento e o proposto por TAN e LE (2019).

Figura 52. Comparação de diferentes métodos de escalonamento. (a) é um exemplo de uma rede padrão; (b) - (d) são escalas convencionais que apenas aumentam uma dimensão da rede largura, profundidade ou resolução. (e) método de escala composta proposto que escala uniformemente todas as três dimensões com uma razão fixa.



Fonte: Adaptado de Tan & Le (2019).

O escalonamento de uma rede pela escala de profundidade (Figura 52 - C) é a forma mais comum de escalonamento. A profundidade pode ser aumentada ou diminuída adicionando e removendo camadas, respectivamente (TAN; LE, 2019).

Já a escala de largura (Figura 52 – B), é comumente usada quando busca se manter o modelo pequeno. Redes mais amplas tendem a ser capazes de capturar recursos mais refinados. Ou seja, aumentar o número de canais nas camadas de convolução - é mais comumente usado para modelos de tamanhos menores (TAN; LE, 2019).

Por fim, a escala de resolução (Figura 52 – D), intuitivamente, devesse dizer que em uma imagem de alta resolução, os recursos são mais refinados e, portanto, as imagens de alta resolução devem funcionar melhor. Mas isso não é escalonado linearmente. O ganho de precisão diminui muito rapidamente (TAN; LE, 2019).

As EfficientNets, por outro lado, executam a Escala composta (Figura 52 – E), ou seja, escalonam todas as três escalas enquanto mantém um equilíbrio entre todas as outras escalas da rede. A ideia principal da Escala composta realmente diferencia a EfficientNets das outras arquiteturas. E intuitivamente, essa ideia de escalonamento composto, faz total sentido, pois, se a imagem de entrada for maior (resolução de entrada), então a rede precisa de mais camadas (profundidade) e mais canais (largura) para capturar padrões mais refinados na imagem maior (TAN; LE, 2019).

A partir dessa arquitetura base, demonstrada no Quadro 16, os autores encontraram os valores equilibrados de escala de profundidade, largura e resolução, utilizando o coeficiente composto fixo. Após essa primeira etapa, tornaram os três valores das escalas fixos e alteraram o valor do coeficiente composto. De modo que, escalonaram a EfficientNet-B0 usando a Escala Composta para obter as arquiteturas EfficientNet B1-B7 (TAN; LE, 2019).

Quadro 16. Arquitetura base EfficientNet-B0 para a ImageNet Challenge

Camada	Tipo	Resolução	Numero de Canais	Numero de camadas
1	Conv 3×3	224×224	32	1
2	MBConv1, $k3 \times 3$	112×112	16	1
3	MBConv6, $k3 \times 3$	112×112	24	2
4	MBConv6, $k5 \times 5$	56×56	40	2
5	MBConv6, $k3 \times 3$	28×28	80	3
6	MBConv6, $k5 \times 5$	14×14	112	3
7	MBConv6, $k5 \times 5$	14×14	192	4
8	MBConv6, $k3 \times 3$	7×7	320	1
9	Conv 1×1 & Pooling & FC	7×7	1280	1

Fonte: Adaptado de TAN & LE (2019).

APÊNDICE B. METRICAS DE AVALIAÇÃO

B.1. MATRIZ DE CONFUSÃO

A matriz de confusão (ou também, tabela de confusão) é uma métrica voltada para modelos de classificação, que tem como objetivo calcular a quantidades de Verdadeiros positivos (VP), Verdadeiros Negativo (VN), Falso Positivos (FP) e Falso Negativos (FN) (HARRISON, 2019).

As duas primeiras, são as classificações corretas, de tal maneira, essas condições assertivas estão na sua diagonal principal. A matriz de confusão de um bom modelo terá a maioria dos exemplos ao longo da sua diagonal principal (GÉRON, 2019).

Uma vez que se obtém a matriz de confusão, todas as outras métricas podem ser obtidas por meio dela. Alguns exemplos são: Acurácia, Precisão, Recall e Especificidade. Assim, podemos entender cada valor presente na matriz de confusão da seguinte forma:

- Verdadeiros positivos (VP): apresenta a quantidade de classificações que foram corretamente classificadas como positivas (SCHADE, 2019), ou seja, o diagnóstico da lesão de pele teve como diagnóstico maligna e realmente era maligna.
- Verdadeiros Negativo (VN): apresenta a quantidade de classificações que foram corretamente classificadas como negativas (SCHADE, 2019), ou seja, o diagnóstico da lesão de pele teve como diagnóstico benigna e realmente era benigna.
- Falso Positivos (FP): apresenta a quantidade de classificações que foram erroneamente classificadas como positivas (SCHADE, 2019), ou seja, o diagnóstico da lesão de pele teve como diagnóstico maligna e realmente era benigna. Esse tipo de erro é conhecido como Erro do tipo I.
- Falso Negativos (FN): apresenta a quantidade de classificações que foram erroneamente classificadas como negativas (SCHADE, 2019), ou seja, o diagnóstico da lesão de pele teve como diagnóstico benigna e realmente era maligna. Esse tipo de erro é conhecido como Erro do tipo II.

Um exemplo de uma matriz de confusão é demonstrado na Figura 53. Em um modelo que auxilia no diagnóstico, devemos sempre minimizar os erros do tipo II, pois, diagnosticar erroneamente algum paciente que não tenha uma lesão maligna e ele seguir para uma biopsia é muito melhor do que diagnosticar um paciente erroneamente que tenha uma lesão maligna e o diagnosticá-lo como uma lesão benigna e o paciente vir a falecer tempos depois.

Uma importante observação, todas as métricas que serão explicadas abaixo, têm suas equações derivadas da matriz de confusão da Figura 53. Além disso, todas as métricas variam no intervalo [0,1] e quanto mais próximos de 1, melhor é o modelo.

Figura 53. Matriz de confusão genérica

		Classe prevista	
		Verdadeiro Positivo (VP)	Falso Negativo (FN) Erro do tipo II
Classe real	Verdadeiro Negativo (VN)		
	Falso Positivo (FP) Erro do tipo I		

Fonte: O autor.

B.2. ACURÁCIA

A acurácia (Equação 18) é uma forma de verificar a proporção de casos que foram corretamente classificados, sejam eles VP ou VN (SCHADE, 2019). Essa métrica avalia o desempenho geral do modelo. De maneira que, devemos utilizar essa métrica somente em modelos que tenham os dados de entrada balanceados. Caso contrário, em casos de dados de entradas desbalanceados, devemos utilizar métricas que observem melhor as quantidades de classificações FP e FN, como o F1-Score e a Acurácia balanceada (BRODERSEN *et al*, 2010).

$$\text{Acurácia} = \frac{VP+VN}{VP+VN+FP+FN} \quad (18)$$

B.2.1. Acurácia balanceada

Como mencionado no parágrafo anterior, a acurácia balanceada difere-se da acurácia pelo fato que ela leva em consideração se os dados de entradas estão ou não balanceados. Caso estejam, os valores das duas acuráncias seriam iguais. Caso contrário, se houver um desbalanceamento, o valor encontrado na acurácia balanceada seria menor do que o da acurácia. A acurácia balanceada, também chamada de *Mean Recall* (Equação 19) é calculada como a média da proporção correta de cada classe individualmente, ou seja, é a soma do Recall mais a Especificidade dividida por dois (BRODERSEN *et al.*, 2010).

$$\text{Acurácia balanceada} = \left[\frac{VP}{VP+FN} + \frac{VN}{VN+FP} \right] / 2 \quad (19)$$

B.3. RECALL (SENSIBILIDADE) E PRECISÃO (VALOR PREDITIVO POSITIVO)

O *Recall* (Equação 20) indica a proporção de casos verdadeiros positivos que foram classificados corretamente, ou seja, entre todas as observações positivas VP e FN (SCHADE, 2019), quantas o modelo conseguiu classificar como verdadeira positiva. De modo que, representa a capacidade do modelo em classificar a classe como verdadeira positiva (GÉRON, 2019).

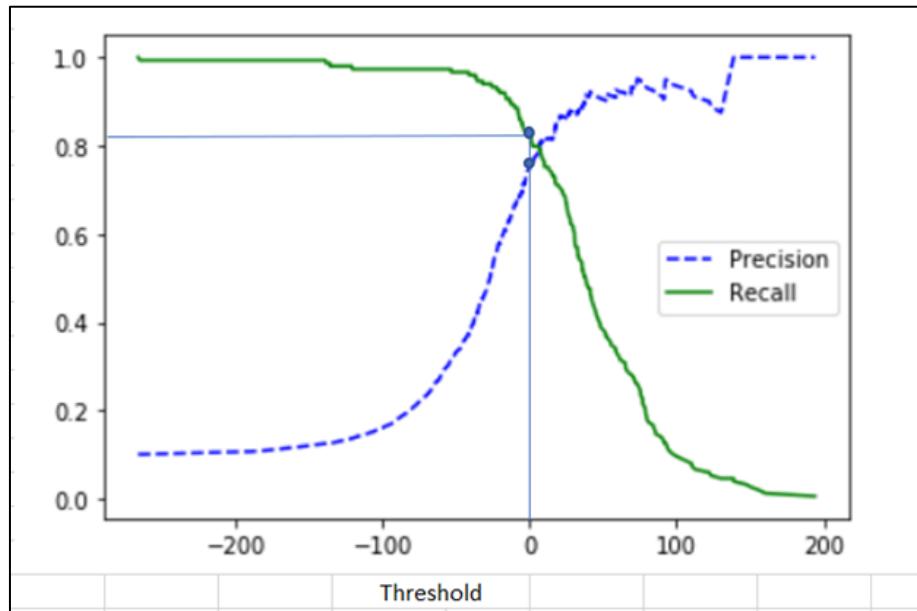
$$\text{Recall} = \frac{VP}{VP+FN} \quad (20)$$

Já a precisão (Equação 21), indica o quão precisas são as classificações positivas (SCHADE, 2019), ou seja, de todas as classificações classificadas como positivas, quantas delas realmente são verdadeiras positivas (GÉRON, 2019).

$$\text{Precisão} = \frac{VP}{VP+FP} \quad (21)$$

Vale mencionar sobre o trade-off Recall-Precisão (Figura 54), em alguns casos devesse priorizar um ou outro. Pois, quando a precisão aumentar, o *recall* deve diminuir e vice-versa (HARRISON, 2019). No caso de modelos que realizam diagnósticos, devesse priorizar o Recall. Para evitar casos errôneos que levam o paciente a óbito, como foi mencionado no tópico da matriz de confusão.

Figura 54. Demonstração do Trade-Off Recall-Precisão

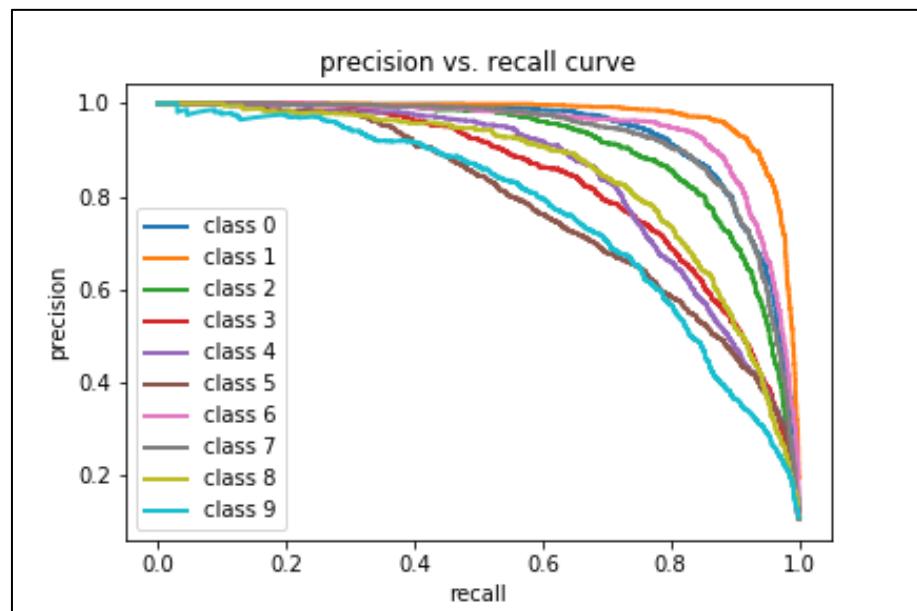


Fonte: Upadhyay (2020).

B.3.1. Curva de precisão-recall

A curva precisão-recall (Figura 55) plota a relação entre os dois, à medida que o limite de decisão é alterado. É um gráfico da precisão (eixo y) e o *recall* (eixo x). O corte que deixa a curva o mais próximo possível do vértice (1,1) irá maximizar a precisão conjuntamente com o recall (AMAZON, 2021).

Figura 55. Exemplo da curva de precisão-recall



Fonte: Miao (2019).

Vale ressaltar que existe a Precisão-Recall AUC, que irá resumir as curvas por meio de valores únicos. De maneira que, será calculada a área sob a curva de precisão-recall. O AUC irá variar entre 0 e 1 e quanto maior o AUC melhor será o modelo.

B.4. F1-SCORE

A métrica F1-Score (Equação 22) é a média harmônica entre o *Recall* e a precisão. Altamente recomenda para avaliar um modelo com dados desbalanceados. A média harmônica é utilizada em vez de uma média aritmética, em virtude da proporção de cada uma (SCHADE, 2019). Para que assim, para o F1-Score ser alto, devesse ter uma precisão e recall altos (GÉRON, 2019).

$$F1 - Score = 2 \frac{Precisão \times Recall}{Precisão + Recall} = \frac{2 \times VP}{2 \times VP + FP + FN} \quad (22)$$

B.5. VALOR PREDITIVO NEGATIVO (VPN)

O VPN (Equação 23) é a métrica que indica a quantidade de classificações negativas que realmente são classificações negativas (SCUDILIO, 2020). Ou seja, dos pacientes que tinham o diagnóstico como benigna, quantos foram diagnosticados como benigna.

$$VPN = \frac{VN}{VN + FN} \quad (23)$$

B.6. ESPECIFICIDADE

A especificidade (Equação 24) é a proporção de casos verdadeiros negativos que foram classificados corretamente. Ou seja, entre todas as observações negativas, quantas o modelo conseguiu classificar como verdadeira negativa. De modo que, representa a capacidade do modelo em classificar a classe como verdadeira negativa (SCUDILIO, 2020).

$$Especificidade = \frac{VN}{VN + FP} \quad (24)$$

B.7. CURVA AUC-ROC

A curva de operação do receptor, também chamada ROC (*Receiver Operating Characteristic*), é a área da taxa de verdadeiro positivo (TVP) versus a taxa de falso positivo (TFP) variando o limiar (VASCONCELLOS, 2018). Essas métricas estão resumidas no Quadro 17. A

curva ROC oferece uma visão geral sobre o modelo, de maneira que, pode ser utilizada para encontrar pontos de corte ideias (GÉRON, 2019).

O corte que deixa a curva o mais próximo possível do vértice (0,1) irá maximizar o *Recall* (Sensibilidade) conjuntamente com a minimização da taxa de falso positivo. É uma ferramenta de diagnóstico popular para classificadores com dados desbalanceados, porque não é tendenciosa entre as classes majoritárias ou minoritárias (CASTRO; BRAGA, 2011).

Quadro 17. Métricas que formam a curva de operação do receptor (ROC - Receiver operating Characteristic)

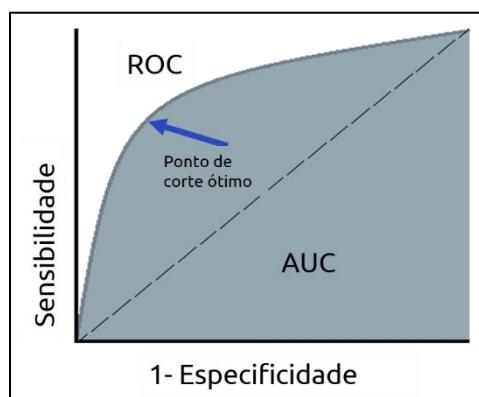
Métrica	Fórmula	Equivalente
Taxa de Verdadeiro Positivo (TVP) (TPR)	$\frac{VP}{VP + FN}$	Recall, sensibilidade
Taxa de Falso Positivo (TFP) (FPR)	$\frac{FP}{VN + FP}$	1-especificidade

Fonte: Adaptado de Amidi & Amidi (2018).

A área sob a curva de operação de recebimento (área em azul), também chamado AUC ou AUC-ROC, é a área abaixo da ROC como demonstrada na Figura 56. O AUC varia entre 0 e 1 e quanto maior o AUC melhor o modelo. A AUC, nada mais é que uma maneira de resumir a curva ROC em um único valor, agregando todos os limiares da ROC (SCUDILIO, 2020).

Na Figura 56 é possível verificar que, quanto mais alto e mais distante da linha diagonal pontilhada a sua curva estiver, melhor. Ou seja, uma curva que se aproxima do canto superior esquerdo do gráfico está se aproximando de um 100% TVP e 0% para TFP, o melhor modelo possível (SCUDILIO, 2020).

Figura 56. Curva AUC-ROC



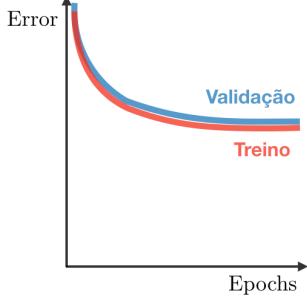
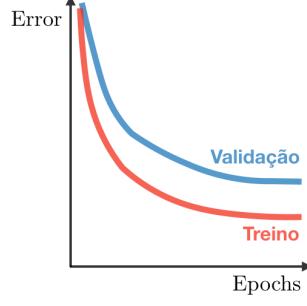
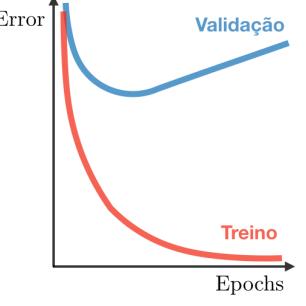
Fonte: Scudilio (2020).

B.8. ANÁLISE DOS MODELOS

Após o treinamento de um modelo, é possível analisar seu erro. De modo, a examinar as classificações errôneas realizadas pelo modelo e assim, entender as causas subjacentes dos erros. Dessa forma, ajudando a priorizar quais dos problemas presentes no modelo, merecem mais atenção.

Por meio do Quadro 18, é possível realizar uma comparação com a plotagem dos gráficos Erro-Epochs. A fim de realizar um diagnóstico do modelo, levando a uma possível direção para lidar com os erros da forma correta e eficiente, ou seja, aplicando o tratamento correto aos erros encontrados (AMIDI; AMIDI, 2018).

Quadro 18. Demonstração dos três tipos de diagnóstico que um modelo pode ter e seus tratamentos

	Sub-ajuste (Underfitting)	Na medida	Sobreajuste (Overfitting)
Sintomas	<ul style="list-style-type: none"> • Erro de treinamento elevado • Erro de treinamento próximo ao erro de validação • Viés elevado 	<ul style="list-style-type: none"> • Erro de treinamento ligeiramente menor que erro de validação 	<ul style="list-style-type: none"> • Erro de treinamento muito baixo. • Erro de treinamento muito menor que erro de validação • Alta variância
Exemplo de Deep Learning			
Possíveis Tratamentos	<ul style="list-style-type: none"> • Modelo de complexificação • Adicionar mais recursos • Treinar mais 		<ul style="list-style-type: none"> • Executar a regularização • Obter mais dados

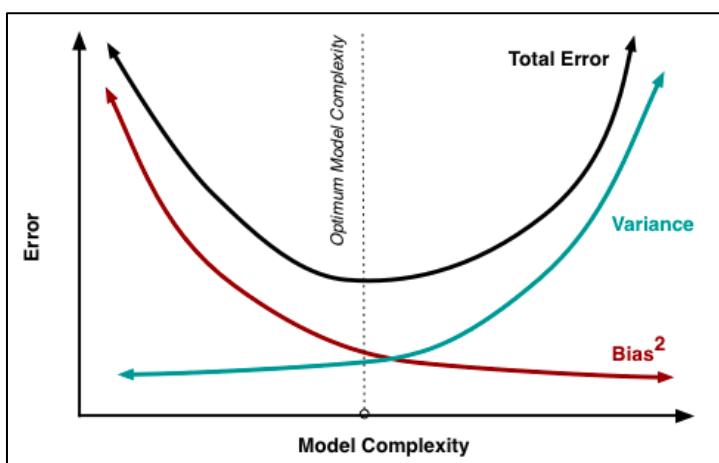
Fonte: Adaptado de Amidi & Amidi (2018).

Ao observar os sintomas do modelo, duas métricas relacionadas a qualidade do modelo são importantes, são elas: viés (*bias*) e variância. Ambas as métricas estão relacionadas com a

capacidade do modelo em classificar futuros dados. O erro de viés está relacionado a incapacidade do modelo em aprender algo sobre os dados, no caso, os dados de treinamento. Já o erro de variância está relacionado a sensibilidade do modelo em aprender com os dados de treinamentos (DUARTE, 2014).

O balanço (*trade-off*) entre viés e variância, apresentado na Figura 57, de forma que, quando se aumenta complexidade de um modelo, o quadrado de seu viés (*bias*) tem seu valor diminuído, enquanto a variância tem seu valor aumentado (HASTIE *et al.*, 2009).

Figura 57 - Trade-Off Viés-Variância



Fonte: Alves (2019).

O erro quadrático médio, que indica a qualidade de um modelo, é a soma da variância e do quadrado do viés. Ele mostra a variação total em torno de um valor verdadeiro (BOLKER, 2008). A seleção de modelos atua na minimização do erro quadrático médio (FORSTER, 2000).

É comum ocorrerem dois tipos de erro devido ao balanço (*trade-off*) entre viés e variância: o sobreajuste(*overfitting*), causado pela alta variância, e subajuste (*underfitting*), causado pelo alto viés (DUARTE, 2014).

O modelo sofre um subajuste, quando não consegue capturar o relacionamento entre os dados de entrada (normalmente denominados X) e suas respectivas soluções (normalmente denominados Y) (AMAZON, 2021).

Já para o sobreajuste, ocorre quando ao se treinar um modelo com os dados de treinamento, o modelo desempenha um ótimo resultado, porém, ao se avaliar o modelo com novos dados, ele apresenta um resultado muito abaixo do anterior. Isso acontece porque o modelo está

memorizando os dados reconhecidos e não consegue fazer a generalização nos exemplos não vistos (AMAZON, 2021).

APÊNDICE C. REVISÃO SISTEMATICA

C.1. QUESTÕES

Para serem selecionados, os trabalhos devem ser capazes de auxiliar na resposta das seguintes questões:

- I. Avaliação da assertividade de um diagnóstico automático de lesões de pele pigmentadas por meio de arquiteturas de redes neurais convolucionais em comparação com dermatologistas.
 - a. P – Pacientes com lesões de pele pigmentadas;
 - b. I – Diagnóstico automático por meio de arquiteturas de CNN;
 - c. R – Diagnóstico realizado por Dermatologistas;
 - d. O – Aumento na assertividade do diagnóstico prematuro de lesões de pele pigmentadas.

- II. Questões complementares:
 - a. QC1: Quais publicações fazem uso de redes neurais convolucionais para classificar automaticamente de lesões de pele?
 - b. QC2: Das publicações que satisfazem QC1, quais publicações classificam automaticamente lesões de pele pigmentadas?
 - c. QC3: Das publicações que satisfazem QC2, quais publicações realizam um pré-processamento das lesões de pele pigmentadas antes da classificação?
 - d. QC4: Dentre as publicações avaliados, quais utilizam *datasets* relacionados ao ISIC como dados de entrada?
 - e. QC5: Satisfazendo a QC4, quais publicações utilizam abordagens para lidar com desbalanceamento dos dados?

 - f.

C.2. REPOSITÓRIO BIBLIOGRÁFICO

III. Dimensions⁵

Esse buscador se trata de um vinculador de informação de última geração, que agrupa diversos repositórios mundiais, e por isso, é capaz de encontrar uma maior quantidade de resultados em diversas fontes. Na data da escrita desse trabalho, é o maior agregador do mundo, com mais de 128 milhões de publicações.

C.3. CRITÉRIOS DE ELEGIBILIDADE

IV. Inclusão:

- a. Publicações relacionados a classificação de lesões de pele utilizando redes neurais convolucionais e suas arquiteturas derivadas;
- b. Publicações relacionados a extração de características de lesões de pele utilizando redes neurais convolucionais e suas arquiteturas derivadas seguidas de um processo de classificação;
- c. Publicações que propõe métodos automatizados de classificação supervisionadas de lesões de pele somente com imagens;
- d. Publicações que utilizem o *dataset* do ISIC 2018 e/ou HAM10000;
- e. A classificação deve acontecer entre as 7 categorias do HAM10000;
- f. Serão apenas aceitas publicações na linguagem português e inglês; e
- g. *Open access* (OA) serão aceitos os tipos: *Hybrid, Gold, Green e Bronze*.

V. Exclusão:

- a. Publicações relacionados a classificação supervisionadas de imagens que não sejam lesões de pele com ou sem a utilização de redes neurais convolucionais;

⁵ <https://app.dimensions.ai/discover/publication>

- b. Publicações que não utilizem métodos de classificação supervisionadas em imagens de lesões de pele;
- c. Que utilizem metadados em conjunto com a classificação supervisionada de imagens;
- d. Publicações com data abaixo de 2017;
- e. Publicações que visem unicamente o processamento de imagens com ênfase ou não em lesões de pele;
- f. Publicações que visem unicamente a segmentação de imagens com ênfase ou não em lesões de pele;
- g. Qualquer material incompleto, ou seja, somente com citação e resumo; e
- h. Não serão aceitos artigos *Closed* dos tipos OA.

C.4. A STRING DE BUSCA

A partir das questões e dos critérios, foi construída a string de busca para encontrar as publicações voltados a satisfazer principalmente a classificação automática de lesões de pele utilizando redes neurais convolucionais e suas diversas arquiteturas.

A string abaixo foi idealizada por meio de uma busca por diferentes publicações relacionadas ao tema por meio da coleta de suas palavras-chaves. Devido às limitações do buscador avançado do repositório bibliográfico, foi realizado uma limitação das palavras utilizadas.

- Buscas em inglês completa: ‘Melanoma’ OR ‘skin melanoma’ OR ‘skin cancer’ OR ‘skin lesion’ OR ‘skin lesions’ OR ‘malignant skin lesion’ OR ‘malignant skin lesion’ OR ‘Malignant tumors’ OR ‘benign lesion’ OR ‘dermatology’ OR ‘dermoscopy’ OR ‘dermoscopy images’ OR ‘dermoscopy image’ OR ‘ISIC’ OR ‘ISIC 2018’ OR ‘HAM10000’ OR ‘Medical diagnostic imaging’ AND ‘deep learning’ OR ‘deep neural networks’ OR ‘convolutional neural network’ OR ‘deep convolutional neural network’ OR ‘image classification’ OR ‘skin lesion classification’ OR ‘Transfer learning’ OR ‘Medical Image Analysis’ OR ‘Medical Decision Support Systems’

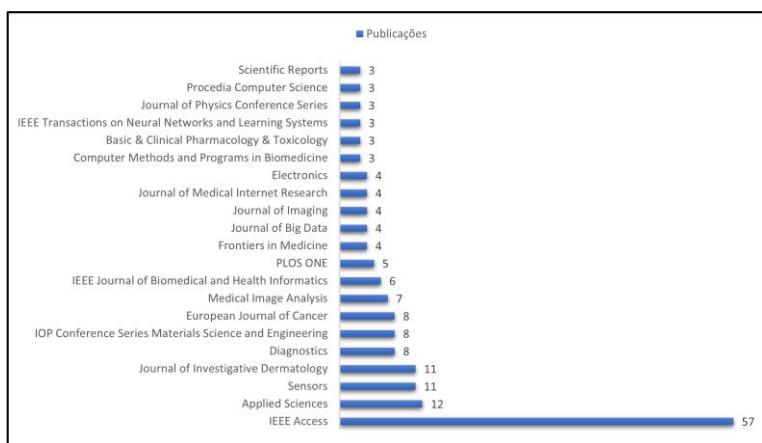
Dessa forma, as duas versões abaixo são as *strings* de busca que foram utilizadas para a obtenção das publicações da revisão sistemática.

- Busca em inglês final: ('Melanoma' OR 'skin cancer' OR 'skin lesion' OR 'dermatoscopic image') AND ('deep learning' OR 'convolutional neural network' OR 'skin lesion classification' OR 'Transfer learning') AND ('ISIC' OR 'HAM10000')
- Buscas em português final: ('Melanoma' OR 'câncer de pele' OR 'lesões de pele' OR 'imagem dermatoscópica) AND ('aprendizado profundo' OR 'rede neural convolucional' OR 'classificação de lesões de pele' OR 'Transferência de Aprendizagem' OR 'Deep Learning' OR 'transfer learning') AND ('ISIC' OR 'HAM10000')

C.5. RESULTADOS

O resultado obtido na busca com base no texto completo dos artigos com o uso da *string de busca* em inglês, já que, em português não houve nenhum resultado correlacionado com o tema desejado. Com a aplicação da *string* no repositório bibliográfico, resultou em 343 publicações, ao qual se dividem entre 151 fontes. Na Figura 58, é apresentado as principais fontes encontradas.

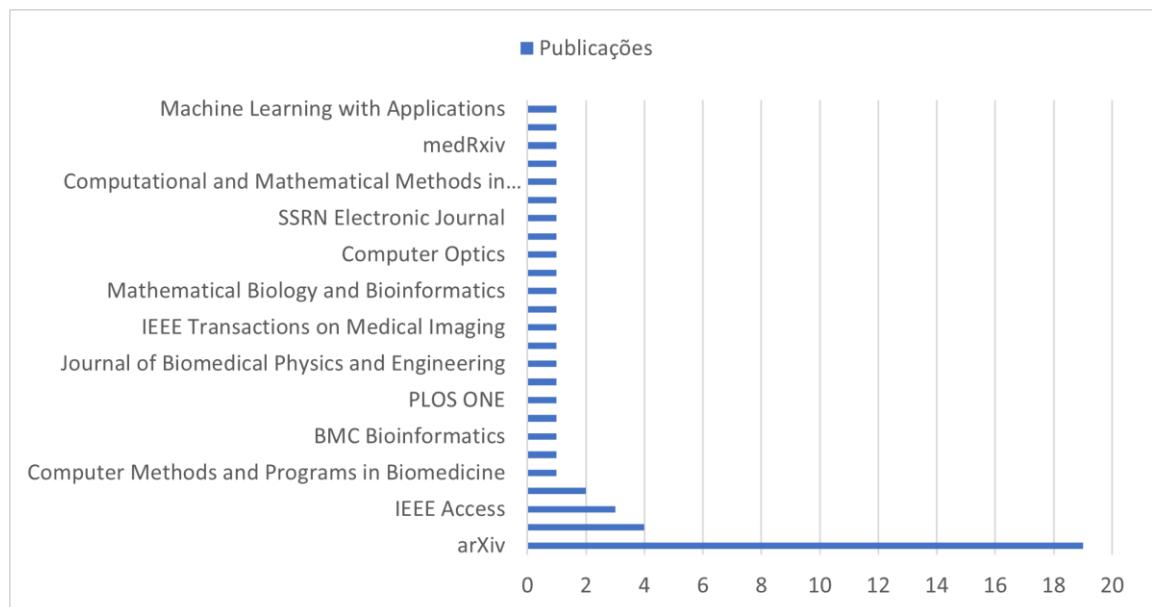
Figura 58. Principais fontes encontradas na busca no texto completo das publicações por meio da string de busca



Fonte: O autor.

Como o montante obtido contém muitas publicações que somente citam as palavras da *string de busca* e não as fazem uso na metodologia. Realizou-se uma nova seleção dessas publicações, com base somente no título e no resumo do artigo. Resultando em um total de 52 publicações. De maneira que, na Figura 59 é apresentado as principais fontes encontradas.

Figura 59. Principais fontes encontradas na busca no título e resumo das publicações por meio da string de busca



Fonte: O autor.

