Classificação Automática para Auxílio no Diagnóstico de Lesões de Pele Usando Deep Convolucional Neural Network com Modelos de Transfer Learning

Derick Abreu Montagna
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Santa Catarina, BR
dam@edu.univali.br

Anita Maria da Rocha Fernandes
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Santa Catarina, BR
anita.fernandes@univali.br

Wemerson Delcio Parreira
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Santa Catarina, BR
parreira@univali.br

Rudimar Luís Scaranto Dazzi
Universidade do Vale do Itajaí - UNIVALI
Itajaí, Santa Catarina, BR
rudimar@univali.br

ABSTRACT

In Brazil, skin cancer has become the most frequent neoplasm among patients. This type of cancer, if detected early, increases the chances of cure. However, with the lack of health professionals qualified to perform this procedure, for example, in distant regions large urban centers, difficulties in early diagnosis process are recurring. Thus, a possible solution to this problem is the development of models that allow classification for the diagnosis skin based on Deep Learning. Therefore, this work presents a model that can contribute to the diagnosis of types of skin devices in the HAM10000 dataset. The proposed model achieved a balanced accuracy of 74.50% with a set.

KEYWORDS

HAM10000, Skin Cancer, Diagnosing, Deep Learning

1 INTRODUÇÃO

A pele é o maior órgão do corpo humano e por conseguinte, está sujeita a lesões. Essas lesões podem ser sinal de doenças graves, como por exemplo, o câncer de pele. O câncer de pele é a neoplasia mais frequente no Brasil segundo [1]. Para diagnosticar o câncer de pele, é necessário a realização de uma biópsia da lesão. Pode-se também utilizar métodos não invasivos complementares, como a dermatoscopia manual ou a dermatoscopia digital. A dermatoscopia é uma técnica não invasiva, no qual o dermatologista usa um aparelho chamado dermatoscópio para fotografar a região da lesão, aumentando-a de 10 a 70 vezes [2].

A detecção precoce do câncer é uma estratégia para encontrar um tumor numa fase inicial e, assim, possibilitar maior chance de cura. Quanto mais precoce for sua identificação, melhores serão os resultados do tratamento [3]. Porém, para a realização dos diagnósticos, atualmente, são necessários profissionais de saúde habilitados para a realização dos procedimentos de diagnóstico das lesões de pele. A grande desigualdade na distribuição da população médica entre regiões, estados, capitais e municípios do interior [4] retarda ou impossibilita o diagnóstico precoce do câncer.

Compreendendo a importância do diagnóstico das lesões de pele em seu estágio precoce, os métodos de processamento digital de imagens e algoritmos de Aprendizado Profundo (do inglês, *Deep Learning* – DL) estão sendo aplicados em 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 [5–7].

Este trabalho busca contribuir com o avanço na área de diagnóstico de lesões de pele com a utilização de inteligência artificial (IA) a partir do desenvolvimento de um modelo baseado em DL. 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. Diante disso, esse trabalho tem como objetivo contribuir com o avanço auxiliar do diagnostico de lesões de pele pigmentadas.

Este artigo está organizado como se segue: Na Seção 2 é apresentada a análise exploratória das imagens do conjunto de dados HAM10000, assim como, o seu pré-processamento, seguido das métricas de avaliação, e por fim, é exposto a linguagem de programação utilizada, os frameworks (bibliotecas) e os ambientes de execução dos experimentos. A Seção 3 apresenta as especificações de como os experimentos foram conduzidos nos três momentos de desenvolvimento dos modelos: treinamento, validação e teste. Na Seção 4, é apresentado os resultados e a discussão a cerca dos modelos desenvolvidos, validados e testados. Finalmente, na Seção 5, é apresentado as considerações finais.

2 MATERIAIS E MÉTODOS

2.1 Análise Exploratória do HAM1000

O conjunto de dados utilizado para este trabalho é o HAM10000 [8], que a partir daqui será referenciado apenas como HAM. O conjunto de dados inclui exemplos representativos de lesões cutâneas pigmentadas que são relevantes na prática. Mais de 95% de todas as lesões diagnosticadas durante a prática clínica cairá em uma das sete categorias de diagnóstico presentes no conjunto de dados [9].

Dessa forma, pode-se separar essas classes em dois grupos: melanocítico e não-melanocítico. Os 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. Após a criação desses dois grandes grupos, define-se a divisão entre as lesões que

são benignas e malignas [10].O resultado dessas separações está representado na Figura 1.

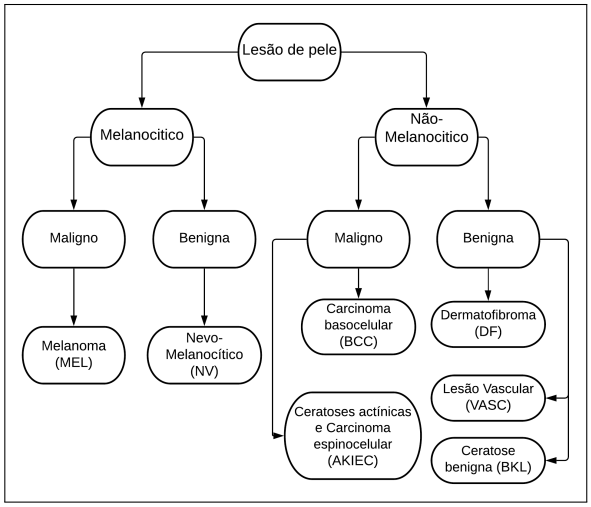


Figura 1: Divisão dos Grupos do HAM10000

A Tabela 1 representa a distribuição dos dados do HAM. Nesta, é possível verificar que as imagens dermatológicas de Nevo melanócito estão desbalanceadas em comparação com as demais categorias. Dessa forma, pode-se verificar que o HAM tem um alto nível de desequilíbrio entre as sete categorias existentes.

Tabela 1: Distribuição dos dados.

Categoria	Quantidade de exemplares HAM10000	Quantidade de exemplares KFCV	Peso Calculado	Peso Utilizado
MEL	1113	1067	4,18	20
NV	6705	5822	0,77	1
BCC	514	479	9,30	10
AKIEC	327	297	15	15
BKL	1099	1011	4,41	5
DF	115	107	41,64	42
VASC	142	129	34,54	35

Devido à distribuição dos dados apresentados na Tabela 1 , 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 às classes minoritárias [11]. Além disso, foi levada em conta, a avaliação do modelo com uma métrica de avaliação que considerando o desbalanceamento, a acurácia balanceada.

O cálculo ocorreu pela divisão do total de exemplares no conjunto de dados de treinamento pelo total de exemplares de cada classe, por fim, o valor foi dividido por 2. 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 classes sofreram arredondamentos para cima. Os valores obtidos podem ser observados na Tabela 1. Os números de exemplares utilizados nos cálculos são da metodologia de validação cruzada em k-partes (do inglês, *K-Fold cross-validation* – KFCV).

Outra importante questão sobre HAM é a presença de imagens únicas. Do total de imagens, 5.515 imagens são únicas e 4.500 imagens tem variantes. Dessa forma, para a criação dos conjuntos de dados de treinamento e validação, todas as imagens foram separadas com base na afiliação da lesão, ou seja, foi assegurado que as imagens da mesma lesão não podem ocorrer em uma divisão de treinamento e validação. Essa separação manual ocorreu devido aos metadados fornecidos pelo autor do conjunto de dados, um arquivo CSV (Valores Separados por Vírgula), contendo: o identificador (id) da lesão, o id da imagem, o diagnóstico, o modo de diagnostico, localização da lesão, e por fim, a idade e sexo dos pacientes [9].

2.2 Pré-processamento das imagens

As imagens presentes no HAM foram adquiridas por múltiplos equipamentos e fontes de iluminações. Dessa maneira, as imagens de mesma classe ou não, apresentam diferentes tons e iluminações. Para tentar minimizar essa diferença, e, ter um ganho de desempenho na classificação, foi aplicado o algoritmo de constância de cor nas imagens médicas. Pois, além desse favorecer a classificação, torna mais justa a comparação entre modelos [12].

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 [12]. Neste trabalho, é utilizado o algoritmo Shades of Grey [13] com a distância de Minkowski Normalizada igual a $6,0$.

Outra técnica aplicada as imagens, foram as de aumento de dados (do inglês, *data augmentation* – DA), de maneira que, é uma técnica capaz de atuar como um regularizador, como também, ajudar a reduzir o *overfitting* ao treinar um modelo de aprendizado de maquina ou aprendizado profundo. Essa técnica, consiste em aumentar a quantidade de dados adicionando cópias modificadas de dados já existentes ou adicionando dados sintético recém-criados a partir de dados existentes [14]. Foram aplicadas as seguintes técnicas de DA no desenvolvimento deste trabalho: Random Flip vertical e horizontal; Random Rotation com fator de 20%, responsável pelo intervalo de rotação; Random Contrast com um fator entre $0,8$ e $1,1$, um limite para a aplicação do contraste na imagem.

Por fim, cada modelo base apresentado, espera um tipo específico de padronização dos dados, sendo que inicialmente os dados estão em uma faixa entre 0 e 255. Como esses valores não são ideais para os modelos bases, é realizada um normalização dos valores, alterando a escala para uma faixa entre 0 e 1. O modelo base, também espera, uma dimensão fixa para as imagens, dessa forma, as imagens foram redimensionadas para 448×448. Assim, os dados passam primeiramente pela camada de redimensionamento, seguido pelas funções de DA, e após, por uma camada de reescalonamento para a padronização.

2.3 Métrica de avaliação

Uma métrica usual para avaliar um modelo com dados desbalanceados é a matriz de confusão (também conhecida como tabela de confusão). Essa métrica é voltada para modelos de classificação, que tem como objetivo calcular a quantidade de Verdadeiros positivos (VP), Verdadeiros Negativos (VN), Falsos Positivos (FP) e Falsos Negativos (FN) [11]. 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 das suas classificações ao longo da sua diagonal principal [15].

Além da matriz de confusão, outra métrica utilizada nesse trabalho, foi o Recall indica a proporção de casos verdadeiros positivos que foram classificados corretamente, ou seja, entre todas as observações positivas VP e FN, quantas o modelo conseguiu classificar como verdadeira positiva. De modo que, representa a capacidade do modelo em classificar a classe como verdadeira positiva [15], dada por:

$$Recall = \frac{VP}{VP + FN}. \tag{1}$$

Já a precisão, indica o quão precisas são as classificações positivas, ou seja, de todas as classificações classificadas como positivas, quantas delas realmente são verdadeiras positivas [15], dada por:

$$Precisão = \frac{VP}{VP + FP}. \tag{2}$$

Vale mencionar sobre o trade-off Recall-Precisão, em alguns casos devesse priorizar um ou outro. Pois, quando a precisão aumentar, o recall deve diminuir e vice-versa [11].

Por fim, a métrica principal desse trabalho, a acurácia balanceada (do inglês, *mean recall* – MR). De maneira geral, se diferencia da acurácia pelo fato que leva em consideração o balanceamento dos dados. Caso estejam, os valores das duas acurácias seriam iguais. Caso contrário, se houver um desbalanceamento, o valor encontrado na acurácia balanceada seria menor do que o da acurácia. MR é calculada como a média da proporção correta de cada classe individualmente [16], dada por:

$$MR = 0,5 \left[\frac{VP}{VP + FN} + \frac{VN}{VN + FP} \right]. \tag{3}$$

2.4 Linguagem de programação, frameworks e ambiente de execução

O `Python` foi empregado como linguagem de programação neste projeto. Todos os experimentos, pré-processamento e arquiteturas que foram avaliadas e implementadas empregando a biblioteca `Tensorflow` com a API `Keras`, como a biblioteca de DL. Além disso, para a visualização dos resultados foi empregado a biblioteca `Matplotlib`. Para as métricas de avaliação utilizou-se a biblioteca `Sklearn`.

Todos os experimentos foram desenvolvidos nas plataformas Kaggle Code e Google Colab Pro no modo GPU, pois o tempo de processamento é muito menor ao treinamento em modo CPU [17].

3 EXPERIMENTOS

Para exemplificar todo o processo de implementação: 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.

3.1 Treinamento

A estratégia de treinamento de um modelo tem demasiada importância no processo de reconhecimento de padrões. Dessa forma, identificou-se que os hiper parâmetros mais relevantes são o ponto de partida *learning rate*, o *learning rate schedule* e a escolha do *early stopping* (primeiro *callback*).

Foi utilizado o otimizador Adam [18], com uma abordagem de taxa de aprendizado chamada Cyclical Learning Rates [19]. De maneira, que foi utilizado o Learning Rate Finder, também, proposto pela mesma autora, para determinar a taxa de aprendizado máxima e mínima utilizadas na técnica citada anteriormente.

O treinamento foi feito em um total de 100 *epochs*, com a chance de parada precoce a cada 15 *epochs*, após as primeiras 30 *epochs*, caso a perda de validação (`val_loss`) do modelo em questão, não apresentasse uma minimização, esse processo tem o nome de Early stopping (primeiro *callback*). Concomitantemente, foi salvo o modelo com a melhor MR no conjunto de validação, com base na avaliação no maior valor de MR de validação a cada *epoch*, esse processo tem o nome de ModelCheckpoint (segundo *callback*).

Como representado pela Figura 2, foi utilizado uma configuração semelhante em cada modelo. Uma camada de *global average pooling*, seguido por uma camada de *dropout* (com uma porcentagem de *drop* de 30%), de maneira que, são adicionadas após a arquitetura base e seguido por uma camada totalmente conectada para previsão. Foi utilizado `Softmax` como a função de ativação na camada de previsão.

Durante o treinamento, cada modelo iniciou-se com o treinamento das camadas do topo, com 30 *epochs*, sem nenhum dos dois *Callbacks*. Posteriormente, foi realizado o treinamento de todo o modelo (camadas do topo e arquitetura base), durante 70 *epochs* ou até o primeiro *Callback* ser ativado. Todos os modelos seguiram as mesmas etapas de DA e sua determinada padronização dos dados. Somente ocorreu a troca do modelo base.

Para calcular a perda, foi utilizada a métrica weight-cross-entropy loss. Foi selecionada como a função de perda, de modo a punir mais duramente as falsas previsões sobre as classes com menos exemplares no conjunto de dados. Com os pesos foram determinados na seção anterior.

As arquiteturas pré-treinadas incluídas neste trabalho como modelos bases são: DenseNet [20], ResNet-V2 [21], InceptionV3 [22], InceptionResNetV2 [23], Xception [24] e EfficientNet [25]. É importante deixar claro que, todos os modelos foram iniciados com os pesos do ImageNet [26]. Dessa forma, utilizou-se o processo de transferência de aprendizado (do inglês, *Transfer learning* – TL), para a afinação dos modelos finais.

Todo o processo de treinamento é condensado na Figura 3. De maneira, que o processo inicia com o HAM10000 com a aplicação

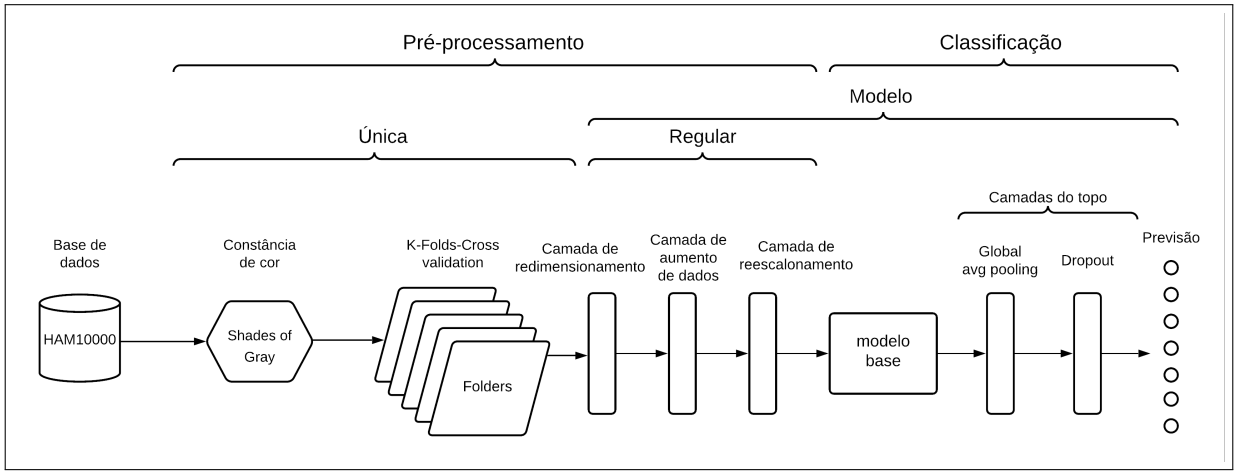


Figura 2: Fluxograma do processo de treinamento

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.

Alguns fatores para a construção dos modelos com um único modelo base são: *(i)* O tamanho da *batch* é 16; *(ii)* O tamanho das imagens de entrada são 448×448; *(iii)* Os passos de treinamento e validação são determinados pela divisão do tamanho do conjunto de dados pelo tamanho da *batch*.

Dessa forma, com as cinco previsões foi possível avaliar os modelos e realizar a média simples entre elas. Por fim, resultando na real avaliação do modelo, terminando assim, o método de validação cruzada em k-partes (KFCV). A métrica de avaliação utilizada nesse processo final do KFCV foi a acurácia balanceada.

3.2 Validação e teste

Na Figura 4, 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.

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.

O subconjunto de teste, contém 1.512 imagens e diferente do subconjunto de validação, não possui uma tabela verdade disponível. Para avaliar a acurácia balanceada, após o processo demostrado na Figura 4, 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 (representada por colunas). Por fim, esses arquivos eram submetidos no ISIC Challenge 2018.3 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.

3.3 Ensemble

Uma técnica simples, mas eficiente para a realização do Ensemble, é calcular a média simples das probabilidades de cada categoria. Portanto, utiliza-se uma média das probabilidades de todos os modelos treinados, dada por:

$$\mu_{ensemble} = \frac{1}{M}({P_{i,j}}^1 + {P_{i,j}}^2 + ... + {P_{i,j}}^M) \quad (4)$$

em que M representa a quantidade de modelos treinados e P a matriz de probabilidades (i representando o número de exemplares e j o número de categorias). Assim, foi desenvolvido um modelo ensemble que continha os seis modelos ($M = 6$) que obtiveram as maiores acurácias balanceadas. Com o $\mu_{ensemble}$ calculado, foi realizado a comparação com os rótulos verdadeiros. De maneira, que a métrica de avaliação utilizada nesse processo foi a mesma do processo de teste, ou seja, a Acurácia balanceada.

4 RESULTADOS E DISCUSSÕES

Os resultados apresentados na Tabela 2 são derivados do processo demostrado nas Figuras 3 e 4. Os valores presentes são dos doze modelos que passaram pelo processo de treino totalizando sessenta treinamentos e doze processos de validação e teste. Os valores estão ordenados da maior para a menor acurácia balanceada.

Como pode-se verificar na Tabela 2 os seis modelos que tiveram o melhor desempenho no treinamento foram: DenseNet169, DenseNet201, EfficientNetB2, EfficientNetB0, DenseNet121 e Inception-ResNetV2. Dessa maneira, que pode-se observar que a maioria dos modelos que alcançaram as acurácias balanceadas mais altas (acima de 76%), não possuem um número tão alto de parâmetros (acima de 15 ilhões). Evidenciando, que o tamanho do modelo base não influência no resultado, mas sim, a forma ao qual a arquitetura está estruturada.

É importante deixar claro alguns pontos relevantes que estão relacionados ao segundo momento de treinamento. O processo de afinamento do modelo – existem quatro formas de proceder

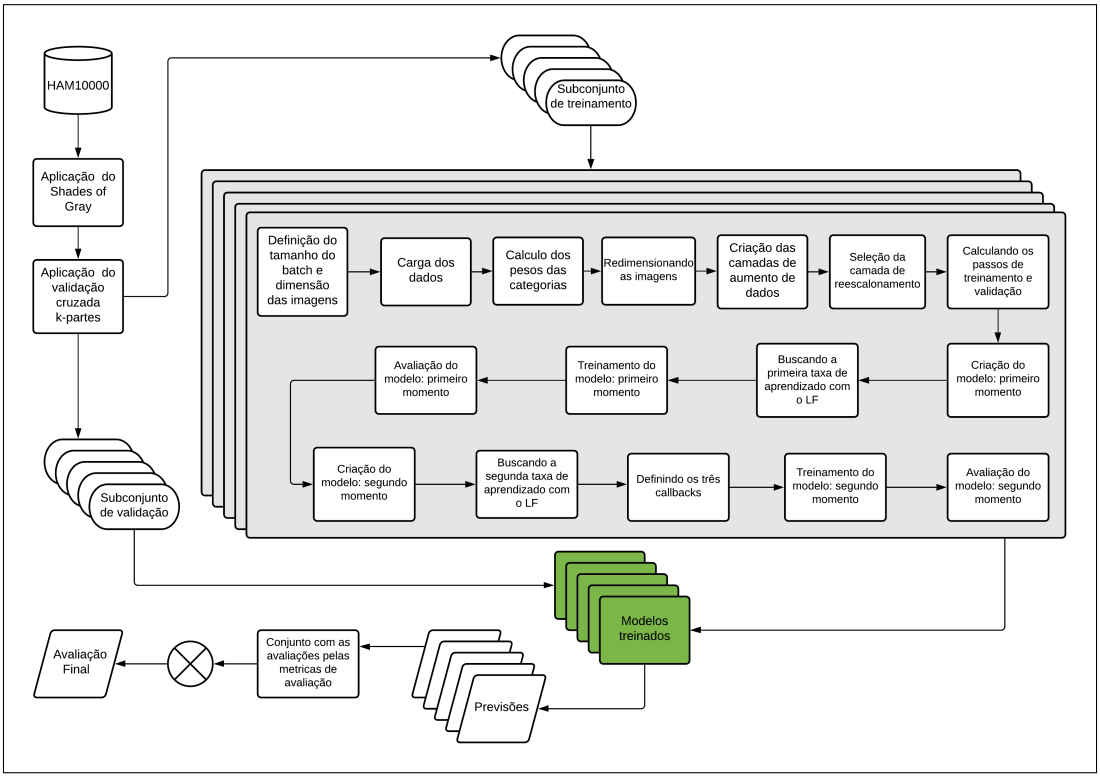


Figura 3: Detalhamento do processo de treinamento

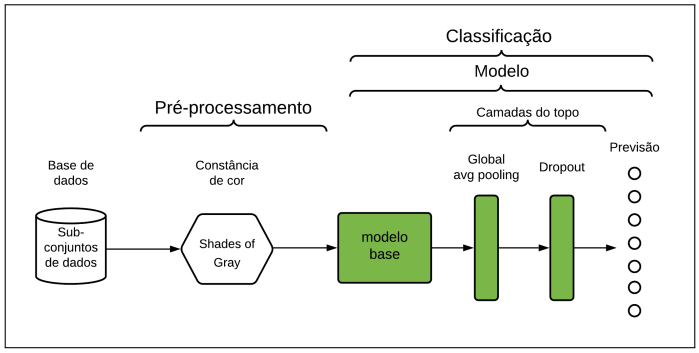


Figura 4: Fluxograma do processo de validação e teste

com o afinamento, levando em consideração que o conjunto de dados utilizado pelos modelos pré-treinados do ImageNet – que é apresentado pela Figura 5 em que cada quadrante é descrito abaixo [27]:

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

Para o desenvolvimento deste trabalho, optou-se por utilizar a opção do Quadrante 1, para que fosse realizado o reaproveitamento do modelo pré-treinado. Isso é devido à dificuldade de encontrar um equilíbrio entre o número de camadas treináveis e congeladas. Uma vez que a falta de equilíbrio causaria um sobreajuste do modelo ao conjunto de dados, ou também, caso fossem congeladas camadas em demasia, o modelo não aprenderia nada útil, ou seja, um subajuste. Outro motivo 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.

Tabela 2: Resultados obtidos com o processo de treinamento com KFCV e os processos de validação e teste.

	Treinamento			Validação	Teste
Modelo	Número de parâmetros	Média de acurácia balanceada (%)	Desvio Padrão	Acurácia balanceada (%)	Acurácia balanceada (%)
DenseNet169	14.307.880	79,8936	2,5901	82,1953	69,80
DenseNet201	20.242.984	79,3066	1,1760	87,6940	70,04
EfficientNetB2	9.177.569	78,1611	2,2560	77,7077	67,20
EfficientNetB0	5.330.571	77,5370	1,1010	77,1069	65,50
DenseNet121	8.062.504	77,0260	4,3044	84,7711	69,99
InceptionResNetV2	55.873.736	76,8825	2,0320	82,9681	68,60
Xception	22.910.480	76,2040	3,7000	78,7395	67,10
EfficientNetB1	7.856.239	76,1550	4,1544	80,2931	66,90
ResNet101V2	44.707.176	75,6358	2,4404	79,5044	68,30
ResNet50V2	25.613.800	75,5308	3,6533	78,3348	66,60
InceptionV3	23.851.784	74,6317	2,6395	76,8764	65,20
ResNet152V2	60.380.648	73,3178	2,7927	67,7359	66,10

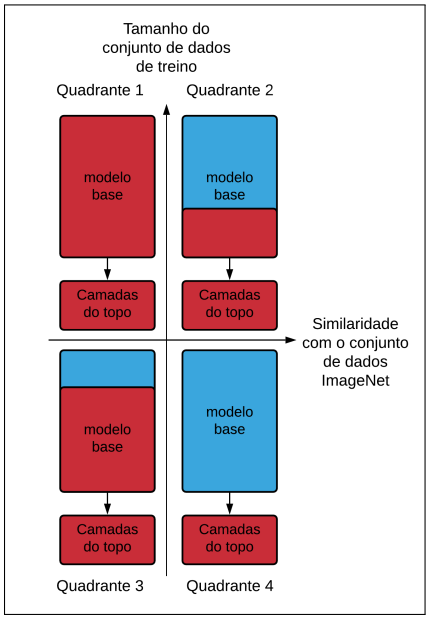


Figura 5: Formas do afinamento

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 em vez de 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.

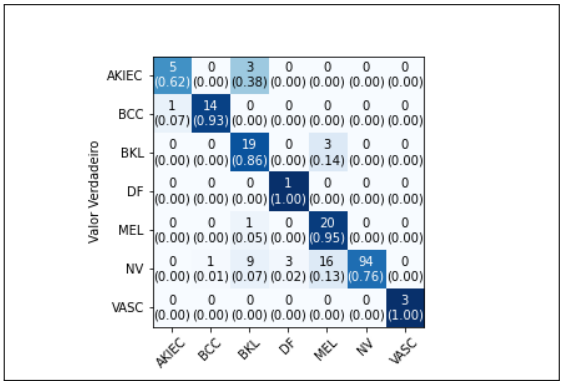
Para o subconjunto de validação e teste, pode-se verificar que os seis modelos que alcançaram a acurácia balanceada mais alta foram (acima de 79% para validação e 66.5% para teste): DenseNet201, DenseNet121, InceptionResNetV2, DenseNet169, EfficientNetB1 e ResNet101V2. Em comparação com os resultados encontrado no processo de treinamento, podemos verificar que dois modelos diferentes entraram no top seis: EfficientNetB1 e ResNet101V2.

Explorando as matrizes de confusão apresentadas na Figura 6, dos oito modelos citados anteriormente verifica-se que, entre as sete categorias, a que teve o menor número de acertos, foi a categoria AKIEC. Mesmo utilizado a entropia cruzada com pesos como a função de perda, que serviu para reduzir o desbalanceamento entre as categorias. Dessa maneira, contribuiu para melhorar o desempenho e superar as categorias majoritárias.

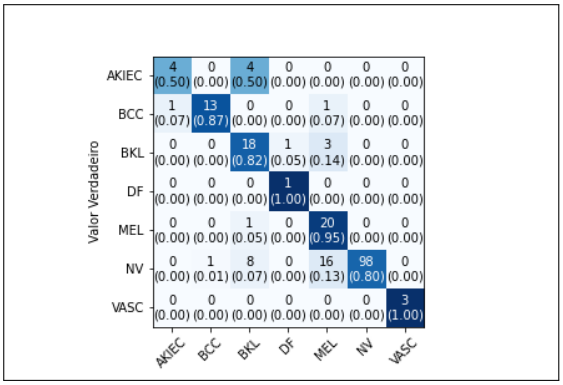
Entretanto, os pesos também podem ponderar uma categoria com uma taxa de verdadeiro positivos mais altas do que as outras categorias. Como ocorreu com DF e VASC. Ademais, as categorias BCC, BKL e NV performaram bem (acima de 70%) de acordo com os pesos estipulados. Por fim, a categoria MEL, que teve 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.

Os dois modelos finais, EfficientNetB2 e EfficienteNetB0, acabaram saindo do top seis no conjunto de validação, devido ao maior número de classificações errôneas em categorias que tinham menos exemplares (como, AKIEC e BCC). Sendo assim, mais severamente penalizadas pela acurácia balanceada, levando a esses dois modelos, a perda das posição entre os seis modelos com melhor desempenho. 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, assim como, a similidade entre algumas categorias.

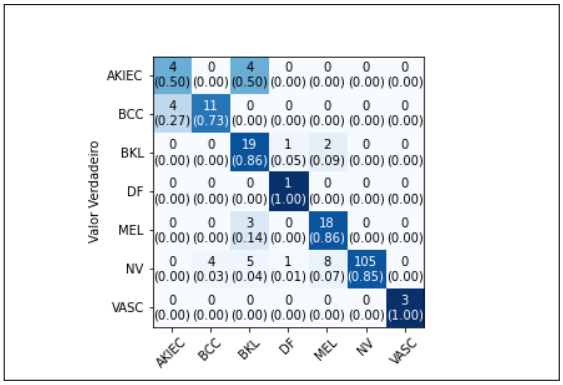
Calculando a precisão-recall, a partir dos valores da Figura 6, todos os modelos apresentam um alto valor de revocação (acima de 70%). No caso de modelos que auxiliam no diagnostico, ter um trade-off para a revocação é um ponto positivo. Pois, deve-se sempre minimizar os erros do tipo II, isto é, diagnosticar erroneamente algum paciente que não tenha uma lesão maligna e ele seguir para uma biopsia é menos danoso ao processo, do que diagnosticar um



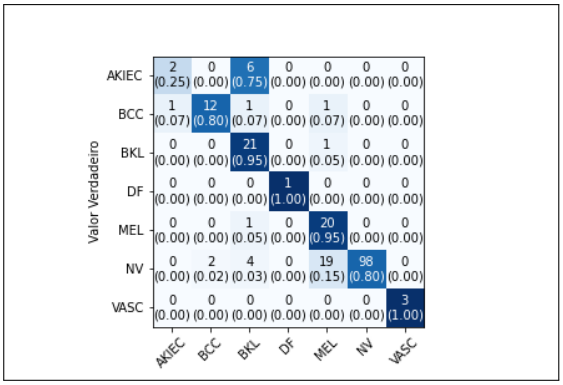
(a) DenseNet201



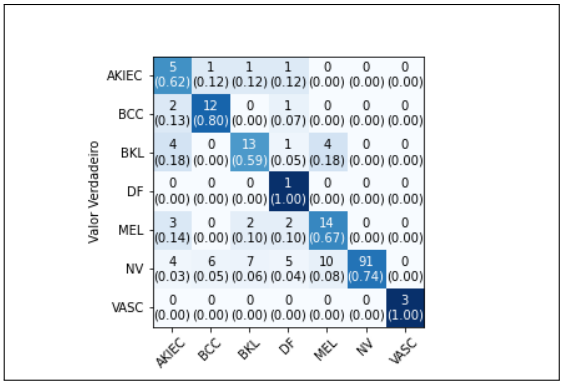
(b) DenseNet121



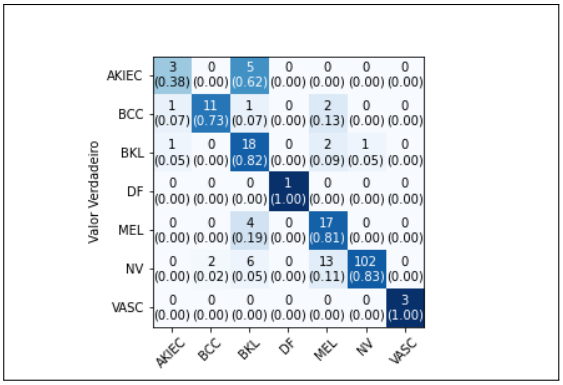
(c) InceptionResNetV2



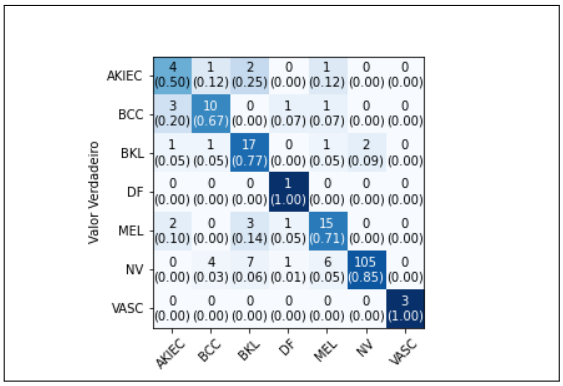
(d) DenseNet169



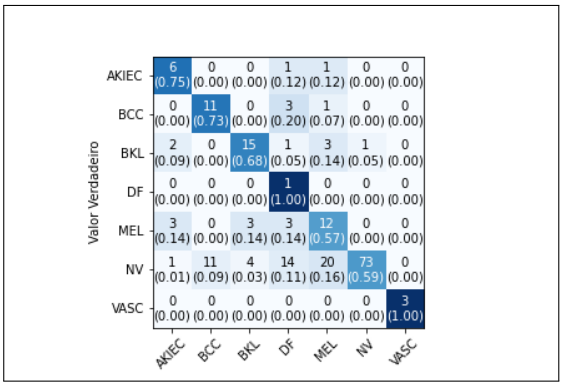
(e) EfficientNetB1



(f) ResNet101V2



(g) EfficientNetB2



(h) EfficientNetB0

Figura 6: Matrizes de confusão de oito modelos da etapa de validação

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.

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 na Seção 3.3. Para os subconjuntos de validação e teste, utilizando os seguintes modelos no ensemble: DenseNet121, DenseNet169, DenseNet201, InceptionResNetV2, EfficientNetB1 e ResNet101V2. Foi possível obter as acurácias balanceada respectivamente para os subconjuntos de validação e teste: 85,25% e 74,5%.

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). Os modelos solos e o modelo ensemble desse trabalho se destacam pelo foco da minimização dos erros do tipo II, ou seja, dos falsos negativos. De maneira, que o valor médio entre os modelos e o modelo ensemble foi de 85% de revocação para esse trabalho.

Outro ponto, a ser mencionado é acerca das estratégias de treinamento. Foram utilizadas estratégias de treinamento com GPU e a realização da carga dos conjuntos de dados na memória ram dos ambientes em nuvem. Dessa forma, reduzindo o tempo de treinamento de cada modelo. Porém, como não houve o afinamento individual de cada modelo pré-treinado, e sim, foram retreinados, não foi possível diminuir ainda mais o tempo de treinamento, variando de 4 a 9 horas dependendo do modelo.

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 evidenciou que a família de modelos pré-treinado DenseNet demostrou resultados com acurácia balanceada média acima de 80% para o conjunto de validação e 69% para o de teste 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.

Todos os códigos e dados utilizados nesse trabalho, podem ser acessados por meio desse repositório.

REFERENCES

[1] M. O. S. Santos. Estimativa/2020 – incidência de câncer no brasil. *Revista Brasileira de Cancerologia*, 66(1), 2020.
[2] G. G. Rezze, B. C. S. de Sá, and R. I. Neves. Atlas de dermatoscopia aplicada. In *Atlas de dermatoscopia aplicada*, pages 190–190. 2004.
[3] J.A.G da Silva. Estimativa 2020: incidência de câncer no brasil. *Rio de Janeiro*, 2020.
[4] M. Scheffer, A. Cassenote, A. Guerra, A.G.A. Guilloux, A.P.D. Brandão, B.A. Miotto, et al. Demografia médica no brasil 2020. *São Paulo: FMUSP, CFM*, 2020.
[5] S. Shen et al. Low-cost and high-performance data augmentation for deep-learning-based skin lesion classification. *arXiv preprint arXiv:2101.02353*, 2021.
[6] P. Yao et al. Single model deep learning on imbalanced small datasets for skin lesion classification. *arXiv preprint arXiv:2102.01284*, 2021.
[7] Soumyya Kanti Datta, Mohammad Abuzar Shaikh, Sargur N Srihari, and Mingchen Gao. Soft attention improves skin cancer classification performance. In *Interpretability of Machine Intelligence in Medical Image Computing, and Topological Data Analysis and Its Applications for Medical Data*, pages 13–23. Springer, 2021.
[8] Papers with code - ham10000 dataset. *Dataset | Papers With Code*. URL https://paperswithcode.com/dataset/ham10000-1.
[9] P. Tschandl, C. Rosendahl, and H. Kittler. The ham10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Scientific data*, 5(1):1–9, 2018.
[10] Nuno Menezes. Dermatoscopia de lesões pigmentadas. *Journal of the Portuguese Society of Dermatology and Venereology*, 69(1):33–48, 2011.
[11] M. Harrison. *Machine Learning – Guia de Referência Rápida: Trabalhando com dados estruturados em Python*. Novatec Editora, 2019. ISBN 9788575228180. URL https://books.google.com.br/books?id=VvXADwAAQBAJ.
[12] C. Barata, M. E. Celebi, and J. S. Marques. Improving dermoscopy image classification using color constancy. *IEEE journal of biomedical and health informatics*, 19(3):1146–1152, 2014.
[13] G. D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *Color and Imaging Conference*, volume 2004, pages 37–41. Society for Imaging Science and Technology, 2004.
[14] C. Shorten and T. M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):1–48, 2019.
[15] A. Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, 2019. ISBN 9781492032595.
[16] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pages 3121–3124. IEEE, 2010.
[17] Y. E. Wang, G-Y. Wei, and D. Brooks. Benchmarking tpu, gpu, and cpu platforms for deep learning. *arXiv preprint arXiv:1907.10701*, 2019.
[18] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
[19] L. N. Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017.
[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.
[21] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.
[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
[23] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
[24] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.
[25] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.
[26] O. Russakovsky et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.
[27] Pedro Marcelino. Transfer learning from pre-trained models, Oct 2018. URL https://towardsdatascience.com/transfer-learning-from-pre-trained-models-f2393f124751.