

Projeto Final: Detecção Histopatológica de Câncer

Fernando dos Reis Santos Filho RA: 234471

Larissa Souto Zagati RA: 178060

Leandro Stival RA:263013

Matheus Neves J O Santos RA: 241827

1 - Resumo

O câncer metastático é o responsável por grande parte das mortes de câncer, assim este trabalho visou desenvolver uma metodologia que, utilizando de imagens de histopatologia, consiga identificar essa categoria da doença. Para isso, foi utilizado diversas arquiteturas de redes neurais convolucionais *CNN* para extrair informações dessas imagens, sendo elas, Xception, EfficientNet B0, EfficientNet B7 e DenseNet 121 onde o resultado da sua extração foi fornecida a uma rede neural profunda *DNN* com o intuito de que essa classifique as imagens entre contendo ou não câncer metastático. Nosso treinamento contou com 10 épocas com o otimizador SGD e a taxa de aprendizado adaptativa.

Nossos resultados demonstraram que as arquiteturas que utilizaram a EfficientNet apresentaram os piores resultados com 50% de acurácia nos dados de teste. Dessa forma, não aprofundamos os nossos esforços nesta arquitetura. Já a Xception e DenseNet 121 se mostraram capazes de gerar bons mapas de características das amostras, visto que suas redes densas obtiveram os melhores resultados, com 85% e 82% de acurácia, respectivamente, para os dados de treino.

Para refinar nossos resultados, foi utilizada a biblioteca SHAP, que permite identificar quais regiões da imagem apresentam um impacto positivo ou negativo na classificação, sendo essa informação representada como uma máscara sobre a imagem original.

Com essa abordagem podemos concluir que ela se mostra eficiente para a detecção de câncer metastático no conjunto de dados e de possível ajuda na análise de diagnósticos médicos, visto que os resultados possuem um certo grau de interpretabilidade.

2 - Introdução

Este trabalho pretende treinar um modelo para identificar câncer metastático em pequenos fragmentos de imagem de varreduras histopatológicas de seções de linfonodos. Estes linfonodos (gânglios linfáticos) são pequenas estruturas que funcionam como filtros para substâncias nocivas, fazendo parte do sistema imunológico. É possível que se desenvolvam dois tipos de cânceres nos linfonodos, o câncer desenvolvido no próprio sistema linfático é chamado linfoma. Porém, devido à sua função no sistema imunológico e sua responsabilidade de drenar substâncias nocivas ao organismo, é mais comum que células cancerígenas provenientes de outros tumores caiam no sistema linfático e se tornem metástases, aumentando os riscos de morte [10]. Assim, no combate ao câncer, é necessário o desenvolvimento de técnicas de classificação de células metastáticas nos linfonodos como indício da presença de tumores em outras regiões do organismo.

3 - Trabalhos Relacionados

A classificação de imagens com câncer metastático também foi abordada no trabalho de Veeling et. al [6] onde utilizando de imagens junto de uma CNN construída utilizando a arquitetura da DenseNet, obteve bons resultados de acurácia 89.8% e AUC 96.3%, a principal técnica descrita no trabalho como de maior contribuição para esses bons resultados foi o pré-processamento dos dados, realizando o aumento das amostras, rotacionando-as e ampliando-as.

A detecção de metástase de pulmão é uma tarefa abordada no trabalho de Liu, Y., et al [7], onde foi implementada uma CNN baseada na arquitetura da Inception Net, gerando assim um novo *framework* para tratar o problema da base de dados Camelyon16 composta por imagens de câncer metastático nos linfonodos. Esse trabalho atingiu o estado da arte para esse conjunto em 2017, com uma taxa de acerto de 92.4% das imagens com tumores.

Utilizar transferência de aprendizado para essa categoria de problema vem se mostrando uma escolha comum entre os pesquisadores, como mostra o trabalho de Mohammad Amin Morid et. al [8], que demonstrou que na última década Xception, DenseNet e VGG vêm sendo as principais arquiteturas para a análise de imagens médicas, sendo que cada uma delas tem atuado mais em regiões específicas do corpo e com formato de imagens diferentes.

Aprendizado de máquina, principalmente o aprendizado profundo (do inglês, *Deep learning*) vem se tornando predominante nas análises médicas como mostra a revisão de SHEN [9], onde são abordados os possíveis motivos dessa escolha. Um dos pontos citados é como essa técnica durante o treinamento perde dados devido às camadas convolucionais, assim conseguem extrair informação e diminuir a dimensão dos dados em simultâneo.

Todos os trabalhos citados são consoante à proposta deste trabalho, demonstrando que a técnica selecionada (transferência de aprendizado com CNN) possui embasamento na literatura para esse problema.

4 - Método

4.1 - Dados

A base de dados é composta por 220.025 imagens RGB com a resolução 32x32 pixels e também está disponível em um arquivo no formato .csv contendo o rótulo de cada imagem, ou seja, se a mesma representa um câncer metastático ou não. Esse dataset está disponível no kaggle [1] desde 2019 e conta com mais 1.000 equipes mesmo após o encerramento da competição. A vantagem da escolha dessa base é que, como ela é um pouco mais antiga, possui diversos tópicos de discussão, o que demonstra a relevância dessa categoria de análise. Outra vantagem dessa base, é que a distribuição das amostras estão consideravelmente balanceadas com uma distribuição de 60/40 entre imagens sem e com metástase e que a variável alvo está bem definida, evitando eventuais problemas no treinamento.

4.2 - Redes Neurais

Para a classificação das imagens, foram utilizadas três arquiteturas de redes neurais convolucionais (CNN), um tipo de rede especificamente arquitetada para processar pixels, por facilitar o reconhecimento de objetos em imagens. Para isso realizou-se em primeiro lugar a separação dos dados entre treino e validação, sendo 60% para treinamento e 40% para validação. O tamanho da base de dados é um dos primeiros problemas para essa técnica. Como trata-se de uma quantidade bastante expressiva de imagens, não é possível alocá-las em memória de uma única vez. Portanto, foi necessário que o treinamento fosse realizado em lote. Cada lote carrega 64 imagens para a memória somente no momento que forem utilizadas para treinamento ou validação.

O framework Keras [2] foi utilizado para implementar as CNNs, que utilizam da transferência de aprendizado, permitindo nos aprofundar no funcionamento das redes, por exemplo, utilizando a rede DenseNet121 [3] como a camada convolucional do trabalho responsável pela extração de características das imagens. Os pesos originais (treinados com a base de dados da ImageNet [4]) foram re-treinados com as imagens do dataset aqui descrito. Ainda sobre a rede DenseNet121, assim como nas demais, os inputs de uma camada mais avançada representam um tipo de “concatenação” dos outputs das camadas anteriores. Desse modo, o surgimento desse tipo de rede foi a partir da necessidade de minimizar o efeito dos “vanishing gradients” em redes com muitas camadas entre as de entrada e saída. A DenseNet121 é uma versão da rede DenseNet que possui cinco *Convolutional* e *Pooling layers*, três *Transition layers* (6, 12, 24), uma *Classification layer* (16) e dois *Denseblocks* (1x1 e 3x3 conv). Desse modo, $5 + (6+12+24+16) * 2 = 121$.

A parte responsável por realizar a classificação das características obtidas pela DenseNet121 é composta por quatro camadas densas utilizando a função de ativação ReLU, inicialização dos pesos utilizando a função *He normal* e um termo de regularização L2 para controlar o overfitting da rede. Entre todas as camadas densas, foi realizada o processo de *batch Normalization* para evitar problemas com o gradiente. Entre as camadas 2 e 3, foi utilizada a técnica de *Dropout* com 20% de chance de zerar os pesos entre os neurônios. Entre a camada 4 e a de decisão foi utilizada a mesma técnica, porém, com a chance de 30%, sendo a camada de decisão uma *Softmax* de duas classes (visto que o problema é binário). Podemos observar uma ilustração das arquiteturas da rede e DenseNet121 na figura 1.

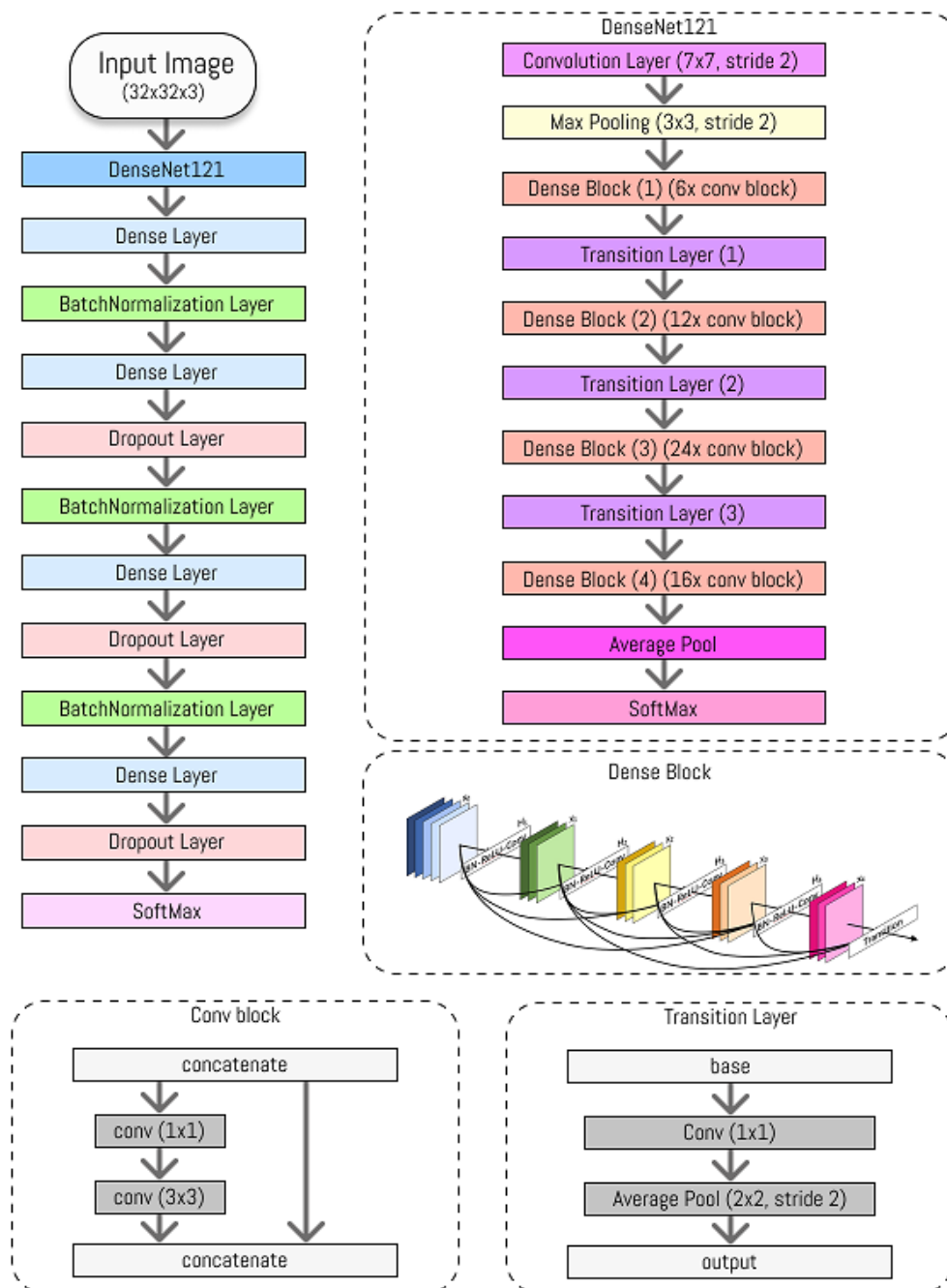


Figura 1: Arquiteturas da rede neural implementada à esquerda e DenseNet121 à direita, com cada bloco detalhado abaixo.

5 - Experimentos

O treinamento foi realizado em 10 épocas utilizando uma GPU para acelerar esse processo, o método de otimização do gradiente utilizado foi a descida do

gradiente com *momentum* (SGD) do Keras, com os parâmetros *momentum* = 0.9 que representa a 'velocidade' de atualização com que taxa de aprendizado é alterada, e com o parâmetro *nesterov* = *True*, modificando, assim, a forma do cálculo do momentum. Ainda foi implementada uma rotina que altera a taxa de aprendizado - definida inicialmente como 0.01 - que vai sendo atualizada a cada interação da seguinte forma:

$$lr = 0.01 * 0.1^{\frac{epoch}{num_epochs}}$$

Onde *epoch* é a época atual e *num_epochs* é o total épocas. O processo de *data augmentation* como SMOTE ou outros não é necessário, visto que nosso conjunto de dados possui uma boa quantidade de amostras.

5.1 DenseNet

É possível observar que o custo (*loss*) de treino começou com um valor elevado e que decaiu rapidamente com o passar das primeiras épocas, indicando uma possível má inicialização dos pesos da rede. Além disso, outro ponto observável na imagem é que a acurácia de validação apresenta uma oscilação. Esse efeito provavelmente pode estar sendo ocasionado pela separação dos dados (lotes com amostras pouco representativas) ou a taxa de aprendizado está dificultando a convergência do modelo.

O processo de treinamento foi significativamente demorado e levou em torno de 5 horas para convergir. A Figura 2 demonstra a evolução do treinamento.

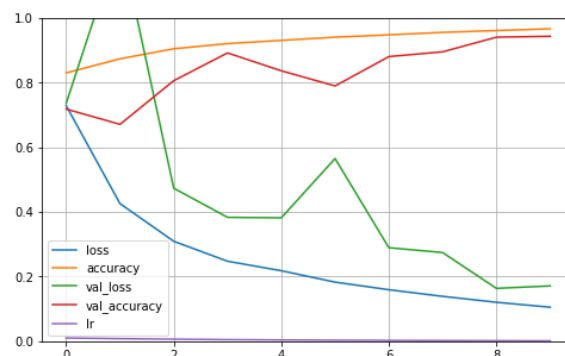


Figura 2: Gráfico da evolução do treinamento da rede em épocas da DenseNet 121.

O maior valor de acurácia no treinamento foi de 99% nos dados de treino e de 94% nos dados de validação. Ao avaliar o modelo nos dados de teste o resultado foi de 82% de acurácia.

5.2 Efficient Net

Os resultados dessa arquitetura se apresentaram estagnados por volta dos 50% e 60% de acurácia, como mostra a Figura 3. Esse processo provavelmente se

deve a diferença entre o conjunto de treino original da rede e as imagens desse trabalho.

Outro motivo pode ser que a quantidade de redes densas não consiga aprender com a saída das redes Efficient Net, onde uma possível alteração nesse parâmetro melhore nosso resultado, ou ainda o tamanho das amostras, dado que seu tamanho é de 96x96x3 e a rede foi originalmente treinada com imagens de 224x224x3. Foram testados diversas abordagens de taxa de aprendizado e otimizador para essa rede, porém, todos mantiveram os resultados abaixo das suas concorrentes, onde a acurácia se manteve estática em 59% durante todo o treinamento, sem alterações na loss. Ao aplicar o modelo aos dados de teste o kaggle informou uma acurácia de 50%.

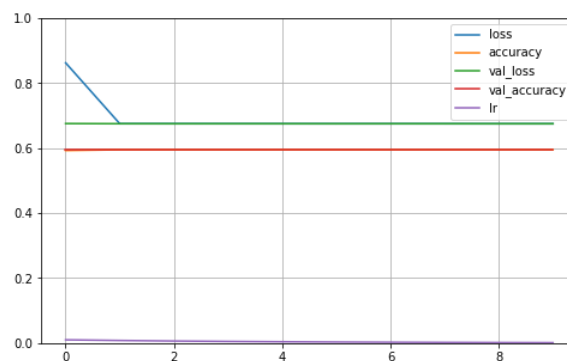


Figura 3: Gráfico da evolução do treinamento da rede em épocas da Efficient Net.

5.3 Xception

A arquitetura apresentou bons resultados para o conjunto de treino como mostra a Figura 4. Isso é condizente com a literatura encontrada, é possível observar algumas oscilações, mas no contexto geral esta arquitetura apresentou bons resultados.

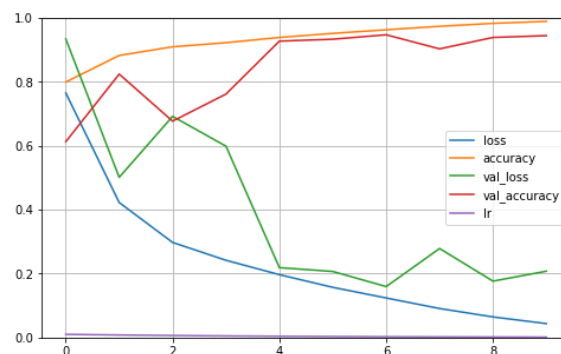


Figura 4: Gráfico da evolução do treinamento da rede em épocas da Xception Net durante o treino.

No kaggle, o resultado para o conjunto de teste foi de 85% de acerto, o que representa uma queda nos resultados, porém ainda demonstra ser uma boa abordagem.

5.4 Importância de cada pixel

Após resolver encontrar os pesos ideais para o modelo para esse problema, pretendemos utilizar alguma técnica para identificar quais pixels das imagens foram de maior impacto na classificação, permitindo identificar as regiões de interesse. Para isso foi utilizada a biblioteca SHAP (SHapley Additive exPlanations) [11], que aplica uma máscara sobre a imagem, de modo a ocultar parte dela para treinar um classificador linear capaz de simular a saída da rede original.

Dessa forma, é possível identificar como cada parte da parte do input influencia na saída do classificador. Em nosso trabalho, realizamos esse processo somente com a melhor arquitetura, ou seja, com a rede com transferência de aprendizado da e ajuste fino da Xception, o resultado dessa técnica pode ser observado na Figura 5.

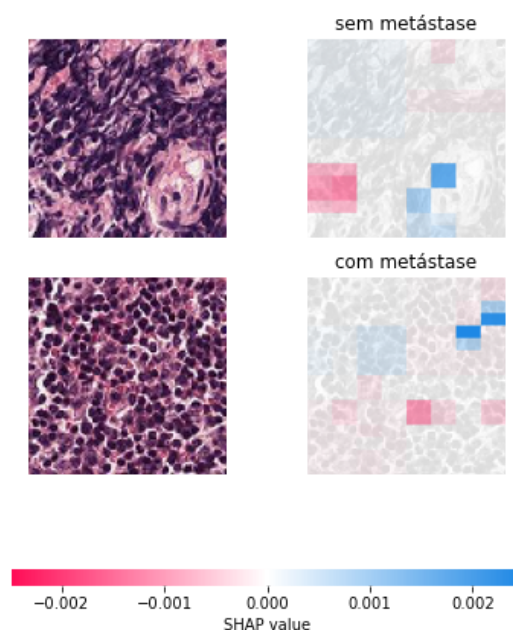


Figura 5: Shap values para duas amostras, uma da classe negativa (sem metástase) e outra positiva (com metástase), onde os tons de azul representam pixels que influenciaram para a rede classificar corretamente a imagem, enquanto os tons de rosa apresentam pixels que indicam que a imagem deveria pertencer à classe oposta.

6 - Conclusão

Das arquiteturas testadas, a que apresentou os melhores resultados, foi aquela que utilizou a transferência de aprendizado da arquitetura Xception. Logo, foi a escolhida como a melhor opção para esse conjunto de dados. Porém, é entendido que o resultado poderia melhorar, caso a quantidade de épocas de treino aumente.

Esse resultado também encontra embasamento na literatura, visto que existem trabalhos de revisão sobre aprendizado de máquina em diagnóstico por imagens, e neles a arquitetura Xception se mostrou bastante presente.

Essa acurácia nos resultados junto da opção de identificar a região da imagem de maior impacto tornam esse projeto uma possível ferramenta para o auxílio nos diagnósticos, dado que o especialista consegue visualizar o porquê daquela amostra ter sido selecionada.

Bibliografia

- [1] Kaggle.com. 2021. Histopathologic Cancer Detection | Kaggle. [online] Available at: <<https://www.kaggle.com/c/histopathologic-cancer-detection/data>> [Acessado 12 de Dezembro de 2021].
- [2] Team, K., 2021. Keras documentation: Keras FAQ. [online] Keras.io. Available at: <<https://keras.io/getting-started/faq/>> [Acessado 12 de Dezembro de 2021].
- [3] G. Huang, Z. Liu, en K. Q. Weinberger, "Densely Connected Convolutional Networks", CoRR, vol abs/1608.06993, 2016.
- [4] Singhal, G. and TensorFlow, I., 2021. Introduction to DenseNet with TensorFlow | Pluralsight. [online] Pluralsight.com. Available at: <<https://www.pluralsight.com/guides/introduction-to-densenet-with-tensorflow>> [Acessado 12 de Dezembro de 2021].
- [5] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.
- [6] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, en M. Welling, "Rotation Equivariant CNNs for Digital Pathology", CoRR, vol abs/1806.03962, 2018.
- [7] Y. Liu et al., "Detecting Cancer Metastases on Gigapixel Pathology Images", CoRR, vol abs/1703.02442, 2017.
- [8] M. A. Morid, A. Borjali, en G. Del Fiol, "A scoping review of transfer learning research on medical image analysis using ImageNet", Computers in biology and medicine, vol 128, bl 104115, 2021.
- [9] SHEN, Dinggang; WU, Guorong; SUK, Heung-Il. Deep learning in medical image analysis. Annual review of biomedical engineering, v. 19, p. 221-248, 2017.
- [10] Mehlen, P., Puisieux, A. Metastasis: a question of life or death. Nat Rev Cancer 6, 449–458 (2006). <https://doi.org/10.1038/nrc1886>
- [11] Shapley sampling values: Strumbelj, Erik, and Igor Kononenko. "Explaining prediction models and individual predictions with feature contributions." Knowledge and information systems 41.3 (2014): 647-665.