

Universidade Estadual de Campinas
Institute of Mathematics, Statistics and Scientific
Computing
(IMECC - Unicamp)

Título do projeto :

Efficient-Unet: Um aprimoramento em modelos
do tipo *U-net*

Orientador: Prof Dr João Batista Florindo
Aluno: Gabriel Borin Macedo

Agosto 2021

Abstract

As tarefas de localização manual de núcleos celulares em imagens de tecido epitelial humano e de mapeamento de vasos sanguíneos da retina são trabalhos que demandam muito tempo e atenção. Com isso, deixando essas localizações altamente suscetíveis a falhas.

Dessa forma, esse projeto investiga o uso de uma rede neural específica denominada *Efficient-Unet* (*Eff-Unet*) para localizar e segmentar núcleos celulares e vasos sanguíneos da retina em imagens médicas. Além disso, é feita uma comparação de desempenho entre os modelos *Eff-Unet* e *U-net* e a respectiva discussão de resultados.

Para ambas as tarefas, a *Eff-Unet* apresentou uma segmentação com maior acurácia. Uma outra característica foi que esse modelo apresentou um treinamento mais rápido e eficiente na tarefa de segmentação de núcleos celulares e uma maior capacidade de generalização na segmentação para a tarefa de vasos sanguíneos. Já o modelo *U-net* apresentou uma maior demora para treinamento na segmentação de núcleos celulares e uma menor precisão e capacidade de generalização na segmentação de vasos sanguíneos. Para o treinamento dos vasos sanguíneos da retina, ambos os modelos demoraram a mesma quantidade de épocas de treinamento.

De modo geral, o modelo *Eff-Unet* demonstrou um grande potencial de aprimoramento dos modelos clássicos de *U-net*, devido a sua fácil escalabilidade, além de um treinamento em menor tempo e com maior acurácia devido à sua capacidade de generalizar mais facilmente. Todo o código e resultado pode ser visualizado **aqui**.

1 Introdução

A crescente popularização de *hardware* com alto desempenho, especialmente de placas gráficas (GPUs), contribuiu fortemente para o aumento de aplicações de redes neurais em diversas áreas da ciência e no dia-a-dia dos hospitais. Dessa forma, possibilitando, por exemplo, avanços significativos na tarefa de localização e de classificação de objetos em imagens. Esses avanços geraram resultados importantes em aplicações como o carro autônomo [2], em sistemas de inteligência artificial que jogam go ou xadrez [8], no auxílio a diagnósticos médicos [3], entre várias outras aplicações.

Em particular, na área de imagens médicas, dois exemplos interessantes estão na utilização desses modelos, principalmente de *U-net* [5] e *Efficient-Unet* [10] em segmentação de núcleos celulares ou até mesmo de vasos sanguíneos da retina. Em especial, o modelo *Efficient-Unet* [1] está ganhando mais notoriedade na literatura e na prática (um exemplo está na incorporação deste modelo à biblioteca *tensorflow* do *python* a partir da versão 2.5.0), devido a sua maior eficiência em menor tempo em relação aos modelos de convolução mais usuais, principalmente comparado com a *U-net*. O trabalho [10] demonstrou que a criação de um modelo que é mais equilibrado em largura, altura e resolução gera

resultados mais precisos em um menor tempo. Outra vantagem apresentada é que esse equilíbrio torna esse modelo escalável em diversos problemas de segmentação. A nomenclatura *Efficient-Unet* deve-se à utilização de uma *Efficient-Net* ao invés de uma *ResNet* como *encoder*

A partir desses resultados descritos na literatura, este projeto de iniciação científica visa investigar o método *Efficient-Unet*, sua performance e acurácia nas tarefas de classificação e localização de núcleos celulares em imagens médicas e de vasos sanguíneos. Além disso, serão comparados esses resultados com os obtidos por uma *U-net* nestas mesmas tarefas e com isso será feita uma discussão para analisar a eficácia de ambos os modelos.

2 Alguns conceitos básicos

Esta sessão apresenta alguns tópicos de grande importância para o entendimento das operações que os modelos *U-net* e *Efficient-Unet* realizam.

2.1 Detecção de objetos utilizando a técnica *Intersection over Union (IoU)*

A técnica de intersecção sobre a união, que também é conhecida na literatura como *Intersection over Union* ou simplesmente por *IoU*, é uma abordagem que permite quantificar se houve sobreposição entre os pixels do objeto previsto pela rede neural e a *grounding truth bounding box* (a qual é a região verdadeira do pixel que pertence ao objeto que queremos classificar)

Para isso, define-se essa métrica da seguinte forma :

$$IoU = \frac{TP}{TP + FP + FN},$$

onde *TP* é o caso da predição correta, *FP* é o caso de falso positivo e *FN* é o caso de falso negativo. Note-se que o valor do *IoU* sempre estará entre 0 e 1.

2.2 Segmentação Semântica

A *segmentação semântica* é uma técnica cujo objetivo é rotular cada *pixel* como pertencente a uma classe de objetos. Ou seja, nessa técnica, cada região da imagem é associada a uma classe.

Diferentemente do método do *IoU*, a *segmentação semântica* realiza a predição dos *pixels* utilizando algumas técnicas de interpolação em vez de regiões retangulares. Dessa forma, o resultado fica com uma melhor precisão e o contorno dos objetos fica melhor delimitado. Entretanto, a desvantagem desta abordagem em comparação com o uso de *IoUs* está em seu custo computacional mais elevado, pois cada *pixel* na imagem precisa ser classificado.

3 Redes neurais do tipo *EfficientNet-Unet* ou *Eff-Unet*

Um problema que redes neurais do tipo *CNN* apresentam é que esse tipo de arquitetura demanda recursos (por exemplo um número pré-determinado de camadas de saltos, camadas customizadas, camadas de convolução, dentre outros exemplos) e que o aumento desses recursos, geralmente, leva a uma melhor performance. Um exemplo disso é que redes neurais do tipo *Resnet18* podem ser aprimoradas para *Resnet50* ou *Resnet101* apenas adicionando mais camadas.

Essa prática envolve um trabalho manual tedioso para implementação desses recursos e que não é garantido produzir um melhor resultado, já que o modelo pode ficar suscetível a problemas de *overfitting*. A partir disso, o trabalho [10] propõe uma família de modelos que escalam sistematicamente e uniformemente a profundidade da rede, largura e resolução das *features* para obter melhores resultados através de um conjunto fixo de recursos. Inicialmente, foi criado o modelo *EfficientNetB0* ou *Eff-NetB0* como um protótipo, e que foi expandido até o modelo *EfficientNetB7* ou *Eff-NetB7*. Dessa forma, esse tipo de *encoder* acaba se tornando aplicável para diversos problemas de segmentação.

3.1 Blocos de convolução *MBConv*

Outro diferencial na *Eff-Net* está na utilização da arquitetura *Mobile Inverted Bottleneck Convolution (MBConv)* [7], que é otimizado para realizar as operações de *squeeze* e *excitation*. Cada modelo baseado na *Eff-Net* possui uma quantidade diferente dos blocos *MBConv*. Além disso, de acordo com [10], a variação *EfficientNetB7* é considerada como o *modelo estado-da-arte* em termos de acurácia na classificação de objetos da *ImageNet*, sendo aproximadamente 8 vezes menor e 6 vezes mais rápido.

3.2 Utilização de uma *U-net* como *decoder*

Modelos *U-net* são redes neurais totalmente convolucionais e simétricas em formato de *U*, que foram originalmente desenvolvidas para segmentação em imagens biológicas. É possível ver mais detalhes sobre o modelo *U-net* no trabalho [5].

As *U-Nets* possuem dois tipos de caminhos. O primeiro é basicamente um conjunto de camadas com convoluções, ativações e *pooling* para capturar *feature maps* da imagem de entrada. Na literatura, é comum se referir a este elemento como *encoder*. Com isso, o resultado dessas camadas, o qual tem dimensão menor do que a imagem inicial, servirá como novo dado de entrada para o próximo caminho do modelo.

Já a outra parte do modelo, também denominado na literatura como *decoder*, realiza uma localização do objeto de forma mais precisa através das operações de convolução transposta. Além disso, o *decoder* combina *features* de alto nível

e informações do espaço da imagem utilizando sequências de operações de *up-convolutions* e concatenação com as suas respectivas *feature maps* provenientes dos *encoders*. Outro fator que se torna favorável para esse modelo está na mistura dos *feature maps* dos *encoders* e *decoders*, a qual permite uma análise mais precisa para identificar múltiplos objetos de interesse em imagens que contêm muita informação.

Para este trabalho, que se baseia na *Eff-Unet*, será utilizado o modelo *Efficient-Net* como *encoder* e uma *Unet* como *decoder*. Inicialmente, o *feature map* final do *encoder* sofre uma operação matemática de *logit*, que será aplicada em um bloco de *up-sample* bilinear por uma escala de dois e por fim será concatenado com o *feature map* do *encoder* que possui a mesma dimensão. Esse processo é repetido até que o mapa da segmentação seja igual ao tamanho da imagem de entrada do modelo. É importante ressaltar que as *Eff-Unets* são assimétricas, diferentemente das *U-nets*, que são simétricas. A utilização de uma *Efficient-Net* como *encoder* aprimora a performance geral do algoritmo, como descrito no trabalho [1]

4 Apresentação e discussão dos resultados práticos

Para avaliação de performance dos modelos, este trabalho adotou o uso de dois conjuntos de dados. A escolha dessas bases é referente ao grau de dificuldade de segmentação dos dados de entrada. Por fim, todos os modelos apresentam *Resnet50* ou *Efficient-NetB0* como *encoder* e uma *U-net* como *decoder*. Por fim, os modelos foram treinados utilizando a biblioteca *tensorflow* versão 2.5.0 em linguagem *python* e foi utilizada uma *GPU NVIDIA GTX 1070* de 8 GB para agilizar no treinamento dos modelos, devido a sua grande capacidade de processamento paralelo.

4.1 Conjunto de dados de segmentação de núcleos celulares

Para esta tarefa, ambos os modelos recebem como dado de entrada imagens médicas em padrão de cor RGB, as quais contêm núcleos celulares. Os modelos devem prever as respectivas máscaras binárias de todas as células. O conjunto de treinamento utilizado é a base de domínio público *2018 Data Science Bowl*, disponível para *download* no site *kaggle.com*.

Um dos desafios dessa base está no pré-processamento das máscaras binárias do treinamento, pois cada imagem de treinamento possui máscaras binárias para cada célula ao invés de uma única máscara binária que contém todos os núcleos segmentados. Por exemplo, se uma imagem contém 20 núcleos celulares, então teríamos 20 máscaras binárias no total, sendo que cada máscara contém somente uma única célula. Para contornar esse problema, as máscaras binárias das respectivas imagens de treinamento são unidas em uma única imagem que contém todos os núcleos.

Outro pré-processamento realizado foi redimensionar as imagens dos dados de entrada de 256×256 para 128×128 . Essa medida foi escolhida pois essa

dimensão na prática demonstrou um resultado mais eficaz em menos tempo computacional. Esse pré-processamento foi adotado para ambos os modelos de *U-net* e *Eff-U-net*.

Além disso, foi implementada uma funcionalidade nos modelos que finaliza o treinamento caso não apresente ganhos significativos na função de perda do teste. Nesse caso, foi definido empiricamente para que o modelo finalize o treinamento caso a função de perda de teste não apresente uma diminuição significativa no valor de perda em 55 épocas de treinamento. Por fim, foi definido que ambos os modelos devam passar por um treinamento de 100 épocas no máximo.

4.1.1 Resultados para a *U-net*

Para o modelo de *U-net*, foram obtidos os seguintes resultados

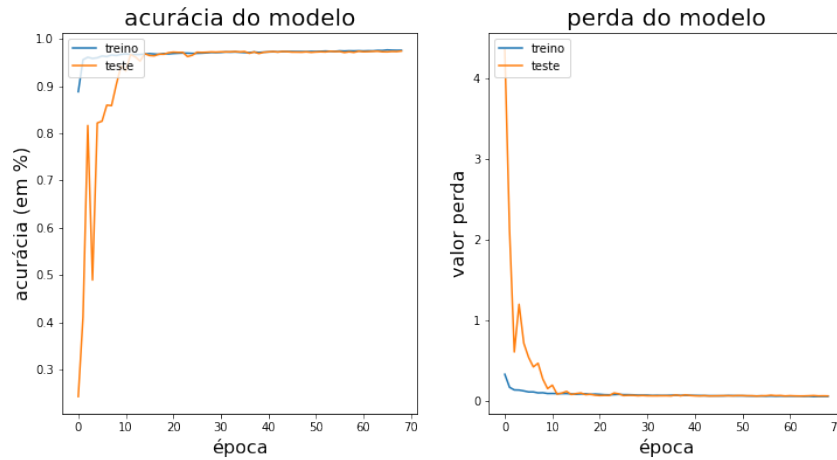


Figure 1: Resultados da função de acurácia e de perda durante o treinamento e teste.

É possível notar-se que o modelo não apresentou problemas durante a fase de treinamento e conseguiu estabilizar sua acurácia em torno da época 15, mantendo-se estável até o fim do treinamento. Esse mesmo argumento também é válido para a função de perda. No final do treinamento, o modelo apresentou precisão em torno de 97%.

Da base toda, 90% das imagens foram utilizadas apenas para o treinamento e o restante apenas para a fase de teste do modelo. A seguir, serão mostrados alguns resultados para as imagens de treinamento, teste e validação. Por fim, o modelo termina com uma perda no treinamento em torno de 0.0580 e de 0.0638

para o teste. Este pode ser considerado um excelente resultado, já que os valores estão próximos entre si e são pequenos.

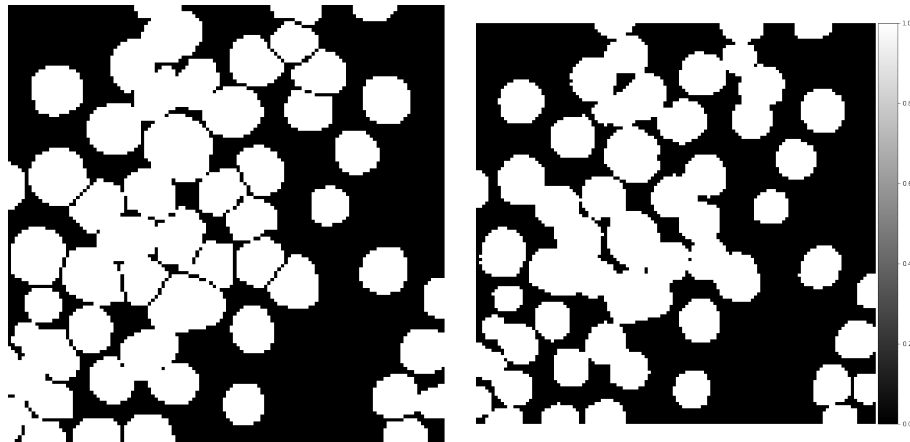


Figure 2: Imagens da base de treinamento, contendo respectivamente a máscara binária original (esquerda) e a máscara binária de saída gerada pelo modelo (direita).

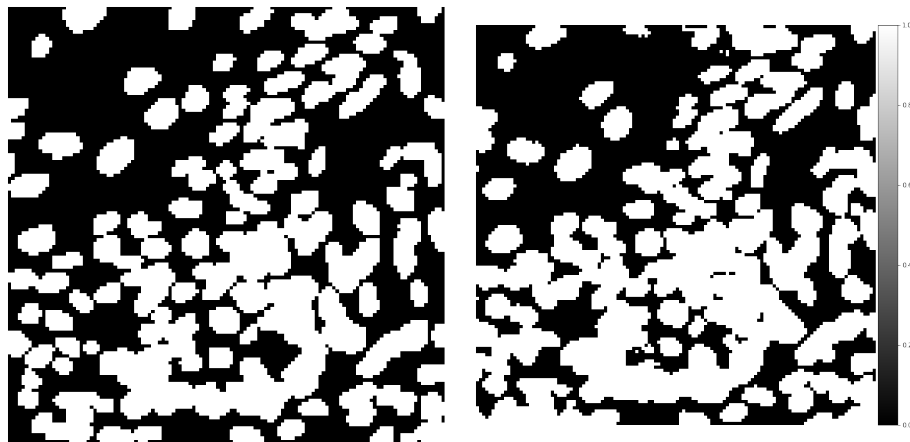


Figure 3: Imagens da base de teste, contendo respectivamente a máscara binária original (esquerda) e a máscara binária de saída gerada pelo modelo (direita).

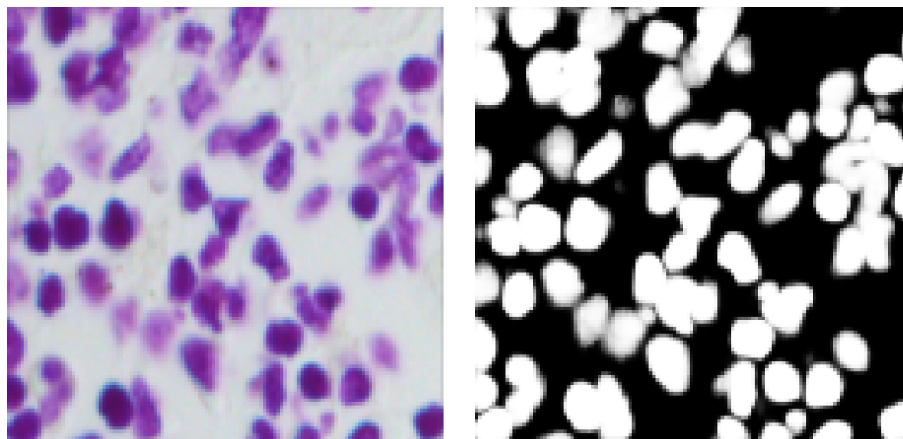


Figure 4: Imagens da base de validação, contendo respectivamente a imagem para segmentação (esquerda) e a máscara binária de saída gerada pelo modelo (direita).

Com esses resultados, é possível notar-se que o modelo realiza uma segmentação com uma excelente taxa de acerto. Entretanto, um problema relevante é que a *U-net* apresenta uma certa dificuldade em segmentar núcleos que estejam próximos. Nesses casos, o modelo aglutina dois núcleos como sendo um só, gerando uma segmentação equivocada. É possível ver a ocorrência desse fato pela imagem de teste, na qual o modelo segmentou várias células como se fossem apenas uma única. Outro problema apresentado é quando há núcleos que estão nos extremos da imagem. Com isso, são gerados buracos nas segmentações de núcleos que deveriam ser contínuos. Um exemplo disso também é visível na imagem de treinamento.

4.1.2 Resultados para a *Eff-UNet*

Para esta fase de teste, foi adotado o modelo *Efficient-NetB0* como *encoder* e foram gerados os seguintes resultados.

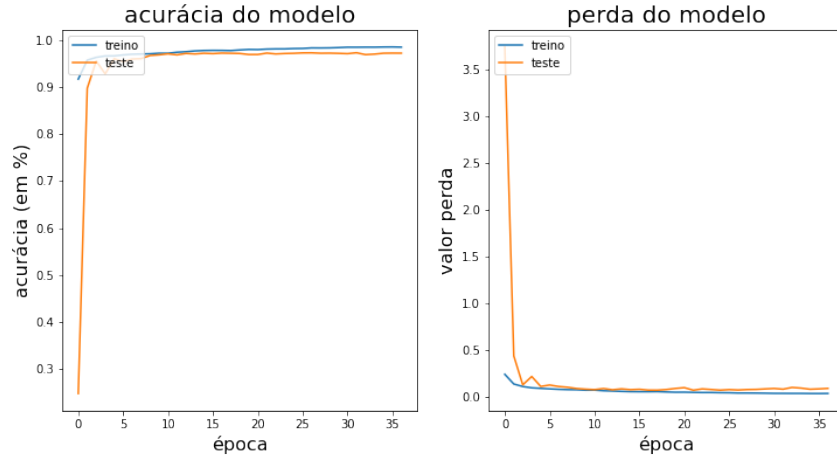


Figure 5: Resultados da função de acurácia e de perda durante o treinamento e teste.

Um fato interessante em comparação com a *U-net* é que a *Eff-Unet* apresentou um treinamento mais estável em menos tempo, tanto para a acurácia quanto para a perda. Outro fator que foi possível notar é que este modelo treinou apenas 36 épocas, enquanto que a *U-net* necessitou de um treinamento de 69 épocas. Nota-se ainda que, a partir da época 11, o modelo não apresentou um menor valor de perda até a época 31, com isso batendo o limite do tempo de treinamento. Por fim, terminou o treinamento com o valor de perda de 0.0370 e de 0.0912 para o teste.

A seguir, são mostrados alguns dos resultados para as imagens de treinamento, teste e validação.

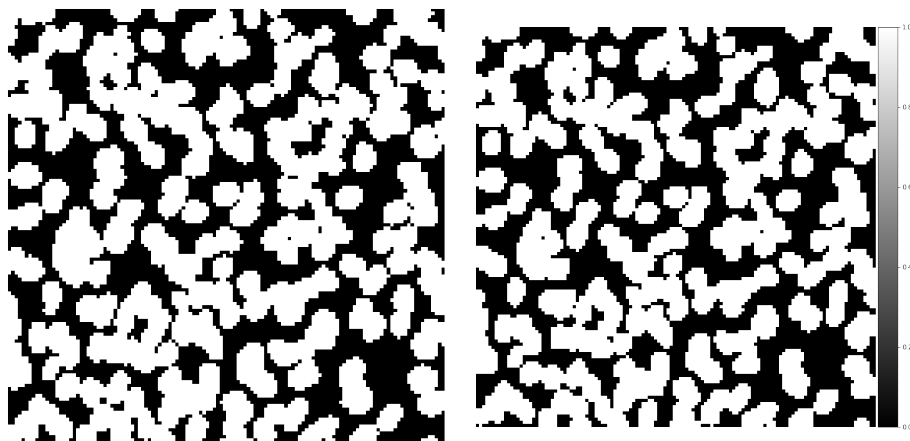


Figure 6: Imagens da base de treinamento contendo respectivamente a máscara binária original (esquerda) e a máscara binária de saída gerada pelo modelo (direita).

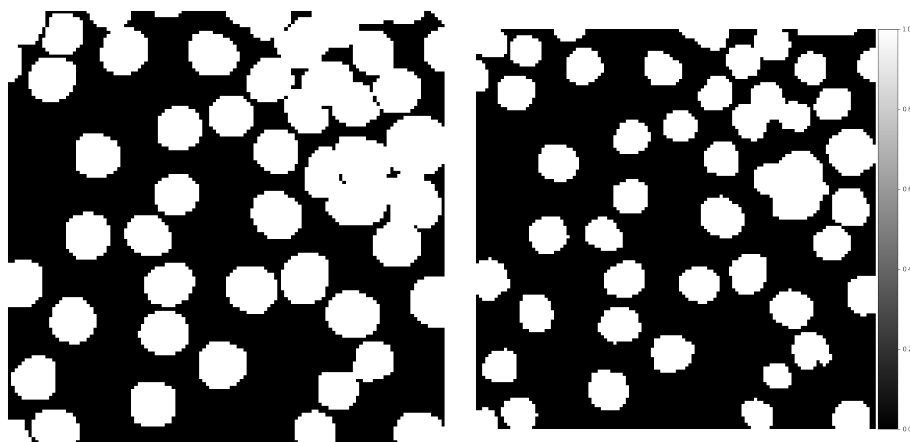


Figure 7: Imagens da base de teste contendo respectivamente a máscara binária original (esquerda) e a máscara binária de saída gerada pelo modelo (direita).

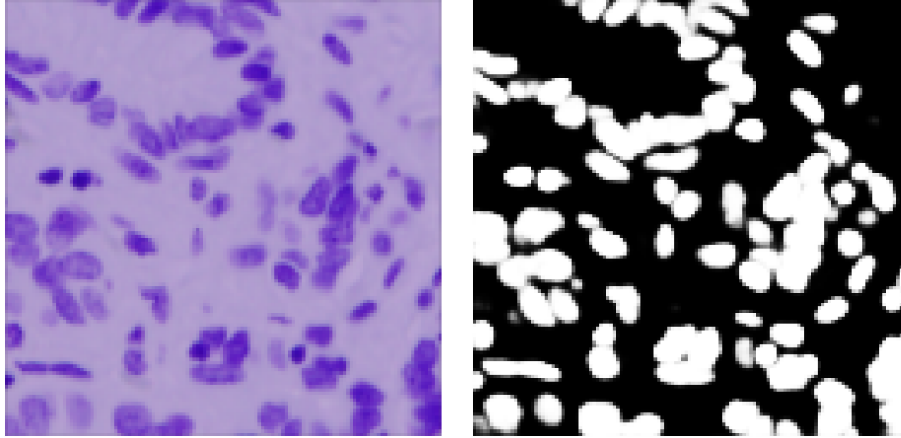


Figure 8: Imagens da base de validação contendo respectivamente a imagem para segmentação (esquerda) e a máscara binária de saída gerada pelo modelo (direita).

De um modo geral, é possível perceber-se que a rede *Eff-Net* realizou uma segmentação mais consistente e precisa em comparação com o modelo anterior. Outra diferença está na maior facilidade da *Eff-Net* em segmentar núcleos que estão próximos ou que estão perto dos extremos da imagem com maior precisão.

4.2 Conjunto de dados de vasos sanguíneos da retina

Para esta parte, os modelos receberão como dado de entrada imagens médicas de retinas, que também estão em padrão RGB e ambos os modelos devem prever as máscaras binárias dos vasos sanguíneos. O conjunto de dados utilizado nesta parte foi a base de dados pública *DRIVE* [9].

O desafio destes dados, além da previsão precisa dos vasos sanguíneos e de suas ramificações, está na limitação da quantidade de imagens. Enquanto que o conjunto de dados de núcleos celulares possui 671 imagens para treinamento, 66 imagens para teste e 3020 imagens para validação, o conjunto *DRIVE* possui apenas 18 imagens para treino, 2 imagens para teste e 20 imagens para validação, deixando o problema ainda mais desafiador pela falta de dados. Inicialmente, foi proposta a utilização de *data augmentation* [4] para criar novas imagens artificiais para suprimir a falta de imagens originais. Entretanto, foi notado com o treinamento que não havia necessidade da aplicação desta técnica, uma vez que um objetivo relevante deste estudo seria exatamente testar o desempenho dos modelos quando há poucas imagens em seu conjunto de treinamento e teste.

Outra modificação feita foi realizar a conversão das imagens de extensão *.tif* dos dados de entrada para o padrão *.png*. O mesmo foi feito para as máscaras binárias que passaram do padrão *.gif* para o padrão *.png*. Essa conversão foi feita para facilitar as operações dos modelos. Além disso, todas as imagens sofreram um redimensionamento de 565×584 para 512×512 , mantendo ainda 3

canais de cores. Por fim, foi definido um treinamento de no máximo 200 épocas para os modelos e um critério de parada caso não houvesse uma diminuição significativa na função de perda do teste em no máximo 55 épocas.

4.2.1 Resultados para a *U-net*

Para esta parte, foram gerados os seguintes resultados de acurácia e perda.

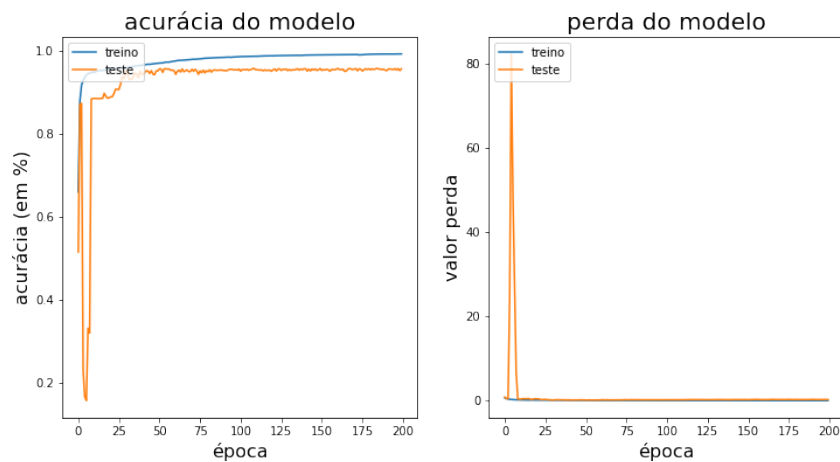


Figure 9: Resultados da função de acurácia e de perda durante o treinamento e teste.

Abaixo, são demonstrado alguns dos resultados gerados pelo modelo.

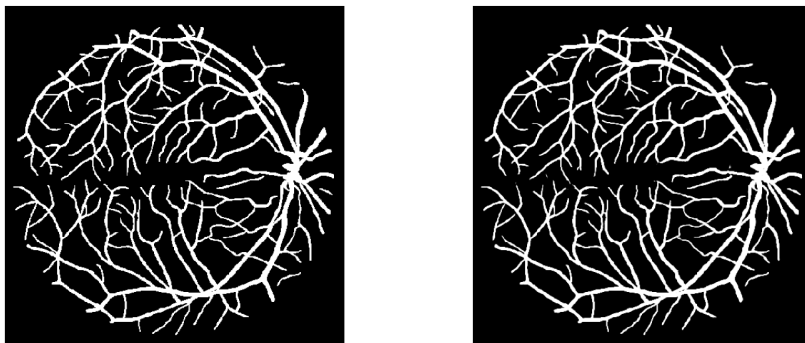


Figure 10: Imagens da base de treinamento contendo respectivamente a máscara binária original (esquerda) e a máscara de saída gerada pelo modelo (direita).

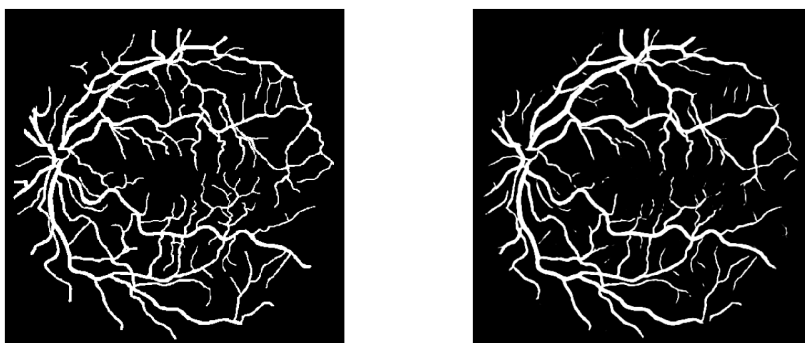


Figure 11: Imagens da base de teste contendo respectivamente a máscara binária original (esquerda) e a máscara de saída gerada pelo modelo (direita).



Figure 12: Imagens da base de validação contendo respectivamente a imagem para segmentação (esquerda) e a máscara de saída gerada pelo modelo (direita).

O modelo conseguiu realizar uma segmentação consistente em grande parte dos vasos sanguíneos, principalmente naqueles que são mais grossos. Entretanto, um problema apresentado foi na segmentação das ramificações de vasos mais finos. Nesse tipo de situação, a *U-net* apresenta falhas na segmentação dos caminhos, de tal forma que não consegue segmentar corretamente essas pequenas ramificações ou acaba criando ramificações que não existem. Um exemplo disso está na imagem de validação acima, na qual a *U-net* em grande parte dos casos não consegue realizar uma segmentação que seja contínua dos pequenos vasos ou até mesmo segmenta vasos que não existem.

Porém, mesmo com esses problemas, ainda obteve precisão de aproximadamente 99% de acurácia no treinamento e aproximadamente 95% de acerto no teste. Outro aspecto que também é visível está na função de perda, a qual finalizou o treinamento com o valor de perda de aproximadamente 0.0152 e de teste de 0.2552, o que é um resultado destoante. Isso na prática pode indicar que o modelo não conseguiu aprender um padrão mais geral para segmentar imagens e consequentemente pode gerar segmentações equivocadas dos vasos.

Uma das explicações para essa diferença provavelmente está nas poucas imagens de treinamento e teste que apresentam mais ramificações nos pequenos vasos, com isso diminuindo a precisão na segmentação. A resolução para esse problema seria treinar novamente esse modelo, porém focando em imagens que apresentem mais essas ramificações. Outra alternativa seria utilizar um outro conjunto de dados para complementar o treinamento do modelo e utilizar os pesos do modelo para treinar nesse novo conjunto de dados. A composição desses dados pode ser obtida tanto de bases públicas quanto utilizando-se dados privados.

4.2.2 Resultados para a *Eff-Unet*

Para a arquitetura *Eff-Unet*, também foi escolhido a *Efficient-NetB0* como *encoder*. Com isso, foram gerados os seguintes resultados de acurácia e de perda

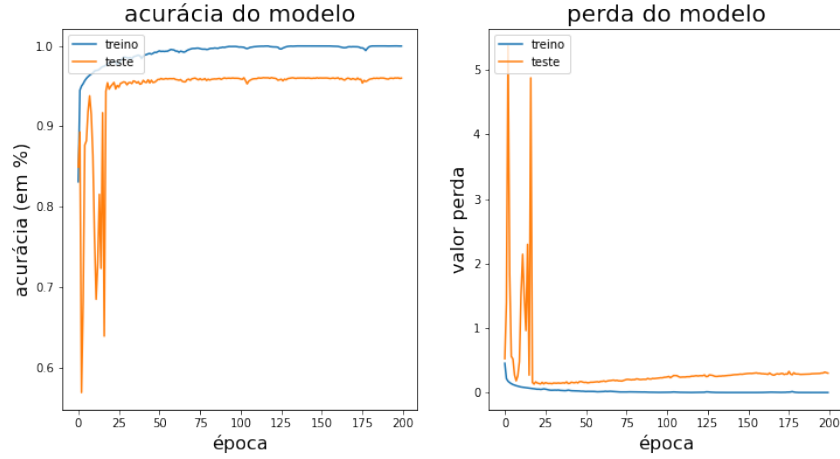


Figure 13: Resultados da função de acurácia e de perda durante o treinamento e teste.

Uma das diferenças desse modelo está na maior variação da acurácia e de perda antes da época 24. Porém, quando a *Eff-Unet* atinge essa época, o modelo se estabiliza até o fim do treinamento. Também é possível notar-se que a perda, mesmo oscilando mais em comparação com a *U-net*, não chegou a 80 em certas épocas e manteve uma tendência de menores valores.

A seguir, podemos visualizar, respectivamente, alguns resultados nos conjuntos de treinamento, teste e validação.

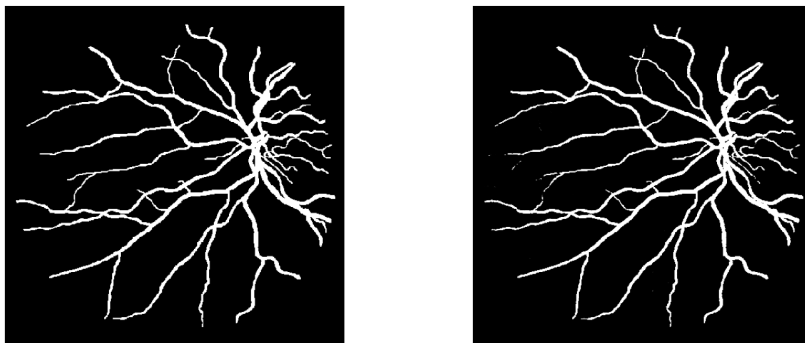


Figure 14: Imagens da base de treinamento contendo respectivamente a máscara binária original (esquerda) e a máscara de saída gerada pelo modelo (direita).

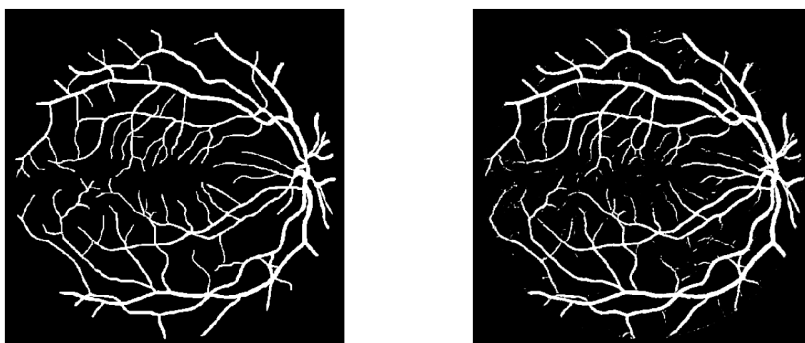


Figure 15: Imagens da base de teste contendo respectivamente a máscara binária original (esquerda) e a máscara de saída gerada pelo modelo (direita).



Figure 16: Imagens da base de validação contendo respectivamente a imagem para segmentação (esquerda) e a máscara de saída gerada pelo modelo (direita).

Com esses resultados, é possível perceber-se que este modelo obteve um ótimo desempenho comparado com o modelo *U-net*. Além disso, obteve uma acurácia de aproximadamente 100% para o treinamento e aproximadamente 96% para o teste. Note que a diferença de acurácia e perda é baixa entre o treinamento e o teste. Além disso, foi notado que o comportamento do treinamento foi similar ao modelo *U-net*, obtendo uma estabilidade em torno da época 24 do treinamento.

Outro fator que podemos perceber é que a perda para a *Eff-Unet* terminou em um valor de aproximadamente $6.4624 * 10^{-4}$ para o treinamento e 0.3011 para o teste, enquanto que o modelo *Unet* terminou com o valor de perda de aproximadamente 0.0152 para o treinamento e de 0.2552 para o teste. Note que neste caso ocorre uma grande discrepância entre os valores de perda da *Eff-Unet*. Uma possível explicação para este fenômeno é que o modelo teve problemas de *overfitting* [6], devido à falta de imagens para o treinamento e de um possível excessivo número de épocas. O modelo *U-net* também apresentou esse problema, porém em menor escala em relação ao valor de perda.

Entretanto, curiosamente a *Eff-Unet* mostrou uma maior capacidade de generalização para segmentação dos vasos sanguíneos. Um exemplo disso está na segmentação da terceira imagem do conjunto de validação.

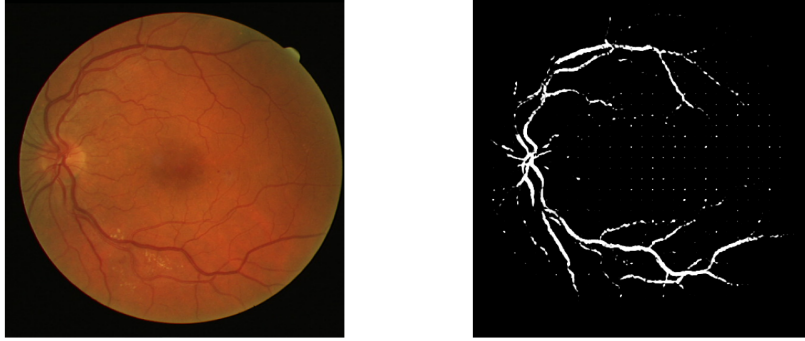


Figure 17: Segmentação da terceira imagem de validação utilizando o modelo *U-net*.

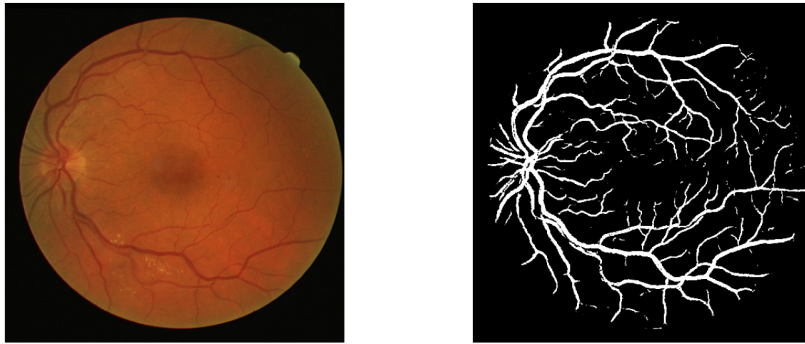


Figure 18: Segmentação da terceira imagem de validação utilizando o modelo *Eff-U-net*.

Na maior parte das segmentações de validação, o modelo *Eff-U-net* também apresentou uma segmentação mais correta em relação ao modelo *U-net*. Com isso, confirma-se a *Eff-U-net* mais uma vez como um modelo mais eficiente para esta tarefa de segmentação. Para visualização de mais resultados, pode-se consultar [aqui](#).

Por fim, o modelo apresentou algumas segmentações de pequenas ramificações que estão equivocadas, principalmente criando ramificações que não existem. Porém, mesmo com esse problema, o modelo ainda apresentou uma segmentação mais correta na comparação com o modelo clássico de *U-net*.

5 Conclusão dos resultados

De modo geral, tanto para a tarefa de segmentação de núcleos celulares quanto de vasos sanguíneos da retina, a *Eff-Unet* apresentou uma segmentação com maior acurácia e confiabilidade em menos épocas de treinamento comparado com o modelo *U-net*. Isso é ainda mais evidente principalmente para a tarefa de vasos sanguíneos, na qual as imagens foram redimensionadas em um formato menor para testar a versatilidade das *Eff-Unet* com imagens de menor tamanho. Mesmo na presença de algumas segmentações equivocadas, principalmente para segmentar pequenos vasos e suas ramificações, o modelo *Eff-Unet* apresentou segmentações mais corretas e uma maior capacidade de generalizar em relação à *U-net*.

Por fim, na tarefa de segmentação de núcleos, ambos os modelos conseguiram um resultado consistente. Entretanto, novamente o modelo *Eff-Unet* apresentou um treinamento mais rápido e com maior precisão em comparação com o modelo *U-net*, tanto na função de acurácia quanto de perda. Outro diferencial é que esse modelo consegue segmentar mais facilmente núcleos que estão próximos ou que estejam nos extremos da imagem.

Com isso, esses resultados reforçam os trabalhos [10] e [1] e sobre a eficácia e desempenho de redes neurais do tipo *Eff-Unet*, sobretudo no caso de modelos do tipo *Eff-Unets*.

References

- [1] BAHETI, B., INNANI, S., GAJRE, S., AND TALBAR, S. Eff-unet: A novel architecture for semantic segmentation in unstructured environment. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (2020), pp. 1473–1481.
- [2] CHEN, B., GONG, C., AND YANG, J. Importance-aware semantic segmentation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems* 20, 1 (2019), 137–148.
- [3] LIU, Y., ZHANG, P., SONG, Q., LI, A., ZHANG, P., AND GUI, Z. Automatic segmentation of cervical nuclei based on deep learning and a conditional random field. *IEEE Access PP* (2018), 1–1.
- [4] PEREZ, L., AND WANG, J. The effectiveness of data augmentation in image classification using deep learning. *CoRR abs/1712.04621* (2017).
- [5] RONNEBERGER, O., FISCHER, P., AND BROX, T. U-net: Convolutional networks for biomedical image segmentation. *CoRR abs/1505.04597* (2015).
- [6] SALMAN, S., AND LIU, X. Overfitting mechanism and avoidance in deep neural networks. *CoRR abs/1901.06566* (2019).
- [7] SANDLER, M., HOWARD, A. G., ZHU, M., ZHMOGINOV, A., AND CHEN, L. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR abs/1801.04381* (2018).
- [8] SILVER, D., HUBERT, T., SCHRITTWIESER, J., ANTONOGLOU, I., LAI, M., GUEZ, A., LANCTOT, M., SIFRE, L., KUMARAN, D., GRAEPEL, T., LILICRAP, T., SIMONYAN, K., AND HASSABIS, D. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* 362, 6419 (2018), 1140–1144.
- [9] STAAL, J., ABRAMOFF, M., NIEMEIJER, M., VIERGEVER, M., AND VAN GINNEKEN, B. Ridge based vessel segmentation in color images of the retina. *IEEE Transactions on Medical Imaging* 23, 4 (2004), 501–509.
- [10] TAN, M., AND LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR abs/1905.11946* (2019).