



**Título do projeto :**

Segmentação semântica usando redes totalmente conectadas baseadas em região: uma aplicação em imagens médicas

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

outubro 2020

# 1 Introdução do projeto e objetivos de pesquisa

A grande disponibilidade de dados que temos atualmente, assim como a relativa popularização de *hardware* com alto desempenho, especialmente placas gráficas (GPUs), tem contribuído fortemente para o crescente número de aplicações de redes neurais em diversas áreas da ciência e da indústria, possibilitando, por exemplo, avanços significativos na tarefa de localização e de classificação de objetos em imagens. Esses avanços possibilitaram a obtenção de resultados importantes em aplicações como o carro autônomo [1], em sistemas de inteligência artificial que jogam go ou xadrez [6], no auxílio a diagnósticos médicos [3], entre várias outras aplicações.

Em particular, na área de imagens médicas, um exemplo de aplicação usando essa técnica é descrita em [3], em que é aplicado um modelo de rede neural para realizar o reconhecimento automático de núcleos de células cervicais. Mais precisamente, foi utilizada uma rede neural denominada *convolutional network per region with mask* (também conhecida como *Mask-RCNN*), juntamente com técnicas de segmentação semântica para reconhecimento de objetos, que neste caso seriam os núcleos dessas células.

## 2 Metodologia matemático-computacional

### 2.1 Redes neurais do tipo *Mask-RCNN*

*Mask-RCNNs* são modelos de redes neurais com o principal objetivo de realizar a tarefa chamada de *instance segmentation*, a qual consiste em utilizar os métodos de detecção de objetos e segmentação em conjunto para identificar objetos e delimitar os *pixels* de uma ou mais classes de objetos, assim como atribuir uma taxa de confiabilidade ao resultado dessa delimitação e identificação. Na prática, modelos de *Mask-RCNN* correspondem à aplicação de dois modelos de redes neurais em paralelo.

Primeiramente, é aplicado um modelo *faster-RCNN* [4] na imagem, o qual é responsável por criar as *bouding boxes* (regiões retangulares que delimitam os objetos de interesse) e classificar essas *bouding boxes*. Toda essa classificação é feita quando são identificadas as *RoIs* nessa imagem.

Em paralelo a esse processo, aplica-se um modelo denominado *Fully Convolutional Network* (FCN) [5] sobre cada um dos *RoIs* encontrados. Assim, a FCN tenta prever máscaras binárias no nível do *pixel*. Dessa forma, esse modelo tem como saída  $C$  imagens binárias com dimensão  $n \times n$  para cada objeto de interesse, em que  $C$  é a quantidade de classes que devem ser identificadas. Por convenção, 1 identifica o *pixel* do objeto e 0 o fundo da imagem. No caso deste projeto,  $C = 1$  pois queremos apenas identificar núcleos em imagens. Essas máscaras serão importantes para delimitar os *pixels* dos objetos encontrados com as *bouding boxes*.

Por fim, o modelo *Mask-RCNN* terá como saída os objetos dentro de *bouding boxes* e delimitados com máscaras com suas respectivas classes, juntamente à

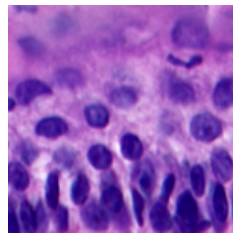
respectiva taxa de confiabilidade do resultado.

## 2.2 Processo final da classificação

Após criar a imagem que contenha as *bounding boxes* e a classificação dos objetos de interesse, aplicam-se as máscaras binárias de cada objeto sobre essa imagem para delimitar os *pixels* dos objetos de interesse, localizados dentro das *bounding boxes*. A grande vantagem desse modelo, além da excelente otimização em seu funcionamento e de sua precisão, está na customização do resultado final. Dessa forma, o usuário pode definir as *bounding boxes* e a qual classe o objeto pertence. Além disso, é possível que se estabeleça uma taxa mínima para a confiabilidade associada ao modelo. No caso deste trabalho, essa confiabilidade mínima foi estipulada em 60%.

## 2.3 Implementação de um novo *branch* para a realização da segmentação

Além do algoritmo dar como saída a imagem descrita anteriormente, esse projeto implementou uma nova metodologia (*branch*) no modelo *mask-RCNN*. Esse procedimento fornece como saída uma outra imagem, que separa os núcleos celulares e deixa o que não é núcleo em preto. Esse processo é bem simples de ser feito. Quando são criadas as máscaras binárias de cada objeto, elas são juntadas em uma única imagem binária que contém todos os objetos de interesse. Após isso, essa junção das máscaras é aplicada em uma cópia da imagem original de entrada do modelo e é comparada *pixel a pixel* em ambas imagens. Se a máscara não contiver o *pixel* do objeto, então o *pixel* será considerado preto na imagem de entrada. O resultado final é visto como a seguir.



(a) entrada de um modelo *Mask-RCNN*



(b) resultado final do novo *branch* adicionado no modelo *mask-RCNN*

Figure 1: Imagem de entrada do modelo e o resultado final do novo *branch* implementado.

### 3 Resultados obtidos

Para a validação do modelo, utilizou-se uma comparação entre as máscaras binárias originais e as geradas pelo modelo. Dessa forma, é criado um arquivo *CSV* durante o treinamento. As colunas correspondem a cada época e cada linha corresponde à porcentagem de acerto da máscara binária final do modelo comparada com a original. No fim das 60 épocas de treinamento, o método *mask-RCNN* apresentou uma precisão em torno de 62%.

Além disso, também são criados os gráficos da função de perda, tanto para a fase de treinamento quanto de validação, utilizando o primeiro conjunto de dados. Os gráficos gerados são exibidos na Figura 2.

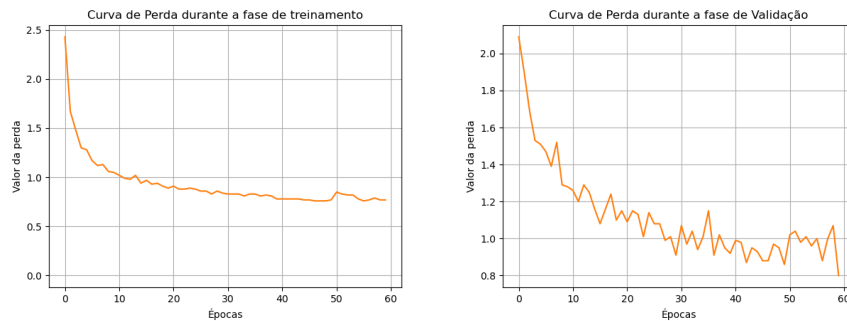


Figure 2: Resultados da função perda durante a fase de treinamento e validação usando o primeiro *dataset*.

Mesmo com o resultado positivo no treinamento, o modelo não obteve bons resultados para classificar e identificar os núcleos presentes nas imagens de validação. Também é possível notar-se que os valores da perda durante a validação oscilaram frequentemente. Essa característica gera uma série de problemas no modelo. Um desses problemas é denominado na literatura como *overfitting*. Neste caso, o modelo aprende pesos ótimos para imagens que apresentam características específicas no treinamento, mas que não necessariamente estão presentes em todas as imagens. Esse tipo de fenômeno faz com que a rede neural fique instável e não seja efetiva na identificação de núcleos em imagens mais gerais.

Entretanto, notou-se que essa rede neural consegue identificar dois núcleos que estão próximos, interligados ou sobrepostos e consegue realizar uma classificação correta em grande parte dos casos. Usualmente, esse tipo de situação é comum e ao mesmo tempo difícil de se tratar. Esse aspecto do modelo é intrigante pois a *mask-RCNN* foi capaz de identificar um padrão complicado utilizando poucos dados. Por fim, foi utilizado o conjunto de dados [2] após o treinamento e validação para verificar o comportamento e o desempenho final do modelo. Mesmo com o problema de *overfitting*, o modelo obteve um desempenho satisfatório e ainda manteve a característica anteriormente descrita.

## References

- [1] B. Chen, C. Gong, and J. Yang. Importance-aware semantic segmentation for autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 20(1):137–148, 2019.
- [2] Joao B. Florindo, Odemir M. Bruno, and Gabriel Landini. Morphological classification of odontogenic keratocysts using bouligand–minkowski fractal descriptors. *Computers in Biology and Medicine*, 81:1 – 10, 2017.
- [3] Yiming Liu, Pengcheng Zhang, Qingche Song, Andi Li, Peng Zhang, and Zhiguo Gui. Automatic segmentation of cervical nuclei based on deep learning and a conditional random field. *IEEE Access*, PP:1–1, 2018.
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. pages 91–99, 2015.
- [5] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(4):640–651, April 2017.
- [6] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dhharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.