

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO TECNOLÓGICO  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA  
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Felipe Sens Bonetto

**REDES NEURAS CONVOLUCIONAIS PARA SEGMENTAÇÃO DE UNIDADES DE  
TECIDOS FUNCIONAIS DO RIM**

Florianópolis

2022

Felipe Sens Bonetto

**REDES NEURAIAS CONVOLUCIONAIS PARA SEGMENTAÇÃO DE UNIDADES DE  
TECIDOS FUNCIONAIS DO RIM**

Trabalho de Conclusão do Curso de Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Danilo Silva, Ph.D

Florianópolis

2022

Ficha de identificação da obra

Bonetto, Felipe REDES NEURAIAS CONVOLUCIONAIS PARA SEGMENTAÇÃO DE UNIDADES DE TECIDOS FUNCIONAIS DO RIM / Felipe Bonetto; orientador, Danilo Silva, 2022. 61 p. Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Santa Catarina, Centro Tecnológico, Graduação em Engenharia Elétrica, Florianópolis, 2022. Inclui referências. 1. Engenharia Elétrica. 2. Inteligência Artificial. 3. Segmentação Semantica. 4. Deep Learning. 5. Tecidos funcionais do rim. I. Silva, Danilo. II. Universidade Federal de Santa Catarina. Graduação em Engenharia Elétrica. III. Título.

Felipe Sens Bonetto

**REDES NEURAIIS CONVOLUCIONAIIS PARA SEGMENTAÇÃO DE  
UNIDADES DE TECIDOS FUNCIONAIIS DO RIM**

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 08 de julho de 2022.

---

Prof. Miguel Moreto, Dr.  
Coordenador do Curso de Graduação em Engenharia Elétrica

**Banca Examinadora:**

---

Prof. Danilo Silva, Ph.D  
Orientador  
Universidade Federal de Santa Catarina

---

Prof<sup>a</sup>. Daniela Ota Hisayasu Suzuki, Dra.  
Universidade Federal de Santa Catarina

---

Prof. Miguel Moreto, Dr.  
Universidade Federal de Santa Catarina

Este trabalho é dedicado aos meus colegas de classe e aos meus queridos pais.

## **AGRADECIMENTOS**

Agradeço aos meus pais por me darem todo o suporte durante toda a minha vida, à minha irmã por ter me guiado pelo mundo espiritual e ao meu orientador por ter me ajudado a realizar este trabalho.

Agradeço aos amigos que fiz até aqui, e a todas as aventuras e desventuras que eu passei para me tornar quem eu sou hoje.

## RESUMO

Redes Neurais Convolucionais pautadas na inteligência artificial e no *deep learning* são agentes facilitadores para resolução de problemas. O objetivo geral deste estudo é treinar um modelo de segmentação semântica de imagens médicas, em várias etapas estruturadas, testando diversos otimizadores, tamanhos de imagem, técnicas de transformação de dados e arquiteturas de redes neurais, buscando a melhor *performance* e a capacidade de segmentação de imagens médicas. Neste sentido apresenta-se a construção de um modelo de inteligência artificial, totalmente convolucional, capaz de segmentar tecidos funcionais do rim, gerando a base tecnológica necessária para auxiliar trabalhos futuros de análise em larga escala dos glomérulos presentes no rim. A relevância se apresenta na contribuição científica em estudos de automatização de exames de diagnósticos para doenças renais e possíveis aplicações para outros ramos. Além disso, ajuda o entendimento de como as interações intercelulares afetam a saúde humana, em específico, na segmentação do glomérulo, um tecido funcional do rim. O modelo final consiste em uma variação da arquitetura U-Net que obteve uma *performance* na métrica DICE de 92,33% na competição *HuBMAP – Hacking the Kidney*, superando 74% dos modelos submetidos à competição.

**Palavras-chave:** Redes Neurais Artificiais Convolucionais. Inteligência Artificial. *Deep Learning*. Tecidos Funcionais dos Rins. HuBMAP. Saúde.

## ABSTRACT

The expansion of research in technology automation is constant at the present time of the fourth industrial revolution. In this scenario, Convolutional Neural Networks based on artificial intelligence and deep learning are facilitating agents for problem solving. The overall objective of this study is to train a model of semantic segmentation of medical images, in several structured stages, seeking the best *performance* and the ability to segment medical images. In this sense, the construction of a fully convolutional artificial intelligence model capable of segmenting functional tissues of the kidney is presented, generating the necessary technological basis to assist future large-scale analysis of the glomeruli present in the kidney. The relevance is presented in the scientific contribution in studies of automation of diagnostic tests for brain diseases and possible applications for other branches. In addition, it helps the understanding of how intercellular interactions affect human health in the segmentation of the glomeruli, a functional tissue of the kidney. The final model consists of a variation of the U-net architecture that achieved 92,33% on the DICE metric on the HuBMAP - Hacking the Kidney competition, performing better than 74% of the models submitted to the competition.

**Keywords:** Convolutional Neural Networks. Artificial Intelligence. Deep Learning. Functional Tissues of the Kidney. HubMAP. Health.



## LISTA DE FIGURAS

Figura 1- Tecido funcional do Rim .....	19
Figura 2 - Redes Totalmente Conectadas .....	15
Figura 3 - Representação de mapa de ativação através da entrada.....	16
Figura 4 - Representação do bloco básico da ResNet .....	17
Figura 5 - Representação da arquitetura da U-Net .....	19
Figura 6 - Representação da arquitetura da U-Net++.....	20
Figura 7 - Camada de Atenção SCSE.....	21
Figura 8 - Imagem transformada com GridDistortion.....	22
Figura 9 - Diagrama de Venn do Coeficiente DICE .....	23
Figura 10 - a) Glomérulo centralizado, b) Glomérulo destacado em vermelho .....	26
Figura 11 - Amostra de imagem de treino .....	26
Figura 12 - Recortes do Córtex, glomérulos destacados em amarelo .....	28
Figura 13 - Recortes da medula.....	28
Figura 14 - Recortes do <i>background</i> .....	28
Figura 15 – Melhor caso obtido com o otimizador <i>ranger</i> , <i>dataset</i> de validação.....	34
Figura 16 – Pior caso obtido com o otimizador <i>ranger</i> , <i>dataset</i> de validação .....	34
Figura 17 - Melhor caso obtido: RMSProp, <i>dataset</i> de validação.....	35
Figura 18 - Pior caso obtido: RMSProp, <i>dataset</i> de validação.....	35
Figura 19 - Curvas de treino e validação: RMSProp.....	36
Figura 20 – Função de perda X Arquitetura .....	37
Figura 21 - Melhor resultado EfficientNet B6, <i>dataset</i> de validação .....	38
Figura 22 - Pior resultado EfficientNet B6, <i>dataset</i> de validação .....	39
Figura 23- Curvas de treino e validação para a Unet-Efficient B6.....	39
Figura 24 – Melhor resultado: Horizontal Flip, Grid Distortion e RandomBrightnessContrast, <i>dataset</i> de validação.....	40
Figura 25 – Pior resultado: Horizontal Flip, Grid Distortion e RandomBrightnessContrast, <i>dataset</i> de validação.....	41
Figura 26 - Curvas de treino e validação do modelo treinado com Horizontal Flip, Grid Distortion e RandomBrightnessContrast .....	41
Figura 27 - Melhor caso: resolução 512x512, <i>dataset</i> de validação .....	43
Figura 28 - Pior caso: Resolução 512x512, <i>dataset</i> de validação .....	43
Figura 29 - Curvas de treino e validação para <i>batchsize</i> =8 e resolução=512x512 .....	44

Figura 30 - Melhor caso do modelo de atenção, <i>dataset</i> de validação .....	45
Figura 31 - Pior caso do modelo de atenção, <i>dataset</i> de validação .....	45
Figura 32 - Amostra totalmente segmentada .....	47
Figura 33 - <i>Leaderboard</i> publico da competição .....	48

## LISTA DE TABELAS

Tabela 1 - Comparação entre U-Net e U-Net++ .....	20
Tabela 2 - Arquitetura de Encoders .....	29
Tabela 3 - Resultado do experimento de otimizadores, <i>dataset</i> de validação .....	33
Tabela 4 - Resultados de diferentes encoders.....	36
Tabela 5 - Resultados do experimento de <i>Data Augmentation</i> , <i>dataset</i> de validação.....	40
Tabela 6 - Resultados para número de épocas diferentes, <i>dataset</i> de validação .....	42
Tabela 7 - Resultados para resoluções diferentes, <i>dataset</i> de validação .....	42
Tabela 8 - Resultados do modelo de atenção, <i>dataset</i> de validação .....	45
Tabela 9 - Resultados U-Net++, <i>dataset</i> de validação .....	46
Tabela 10 - Resultados nas imagens de validação.....	47
Tabela 11 - Comparação de resultados.....	49

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>15</b>
1.1	TRABALHOS RELACIONADOS .....	17
1.2	OBJETIVOS .....	18
<b>1.2.1</b>	<b>Objetivo Geral.....</b>	<b>18</b>
<b>1.2.2</b>	<b>Objetivos Específicos .....</b>	<b>18</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEORICA.....</b>	<b>19</b>
2.1	TECIDOS FUNCIONAIS DO RIM.....	19
2.2	REDES NEURAIAS ARTIFICIAIS .....	20
<b>2.2.1</b>	<b>Redes totalmente conectadas .....</b>	<b>15</b>
<b>2.2.2</b>	<b>Redes Neurais Convolucionais .....</b>	<b>15</b>
<b>2.2.3</b>	<b>Residual Neural Network (ResNet).....</b>	<b>17</b>
<b>2.2.4</b>	<b>EfficientNet.....</b>	<b>18</b>
2.3	Redes de segmentação .....	18
<b>2.3.1</b>	<b>U-Net .....</b>	<b>19</b>
2.3.1.1	<i>U-net++.....</i>	20
2.3.1.2	<i>Camadas de Atenção .....</i>	21
2.4	Data Augmentation.....	22
2.5	MÉtricas de avaliação E Função de perda.....	23
<b>2.5.1</b>	<b>Coeficiente DICE .....</b>	<b>23</b>
<b>2.5.2</b>	<b>Função de perda.....</b>	<b>24</b>
<b>3</b>	<b>Desenvolvimento .....</b>	<b>25</b>
3.1	RESTRICÇÕES COMPUTACIONAIS .....	25
3.2	Conjunto de dados .....	25
<b>3.2.1</b>	<b>Pré processamento .....</b>	<b>27</b>
3.3	Modelos e arquiteturas.....	29

<b>3.3.1</b>	<b>Modelo baseline U-Net .....</b>	<b>29</b>
<b>3.3.2</b>	<b>Variações da U-Net .....</b>	<b>29</b>
3.3.2.1	<i>Modelos de atenção .....</i>	30
3.3.2.2	<i>U-Net ++ .....</i>	31
3.4	Otimizadores.....	31
3.5	Data Augmentation.....	31
3.6	Tamanho do <i>patch</i> .....	32
<b>4</b>	<b>Experimentos e resultados .....</b>	<b>33</b>
4.1	OTIMIZADORES .....	33
4.2	VARIAÇÕES DE ARQUITETURA.....	36
4.3	Data augmentation .....	39
4.4	Tamanho do patch e batch size .....	42
4.5	Modelos de atenção .....	44
4.6	U-Net ++.....	46
4.7	Performance do modelo em imagens inteiras.....	46
4.8	RESULTADOS DA COMPETIÇÃO HUBMAP - HACKING THE KIDNEY...	48
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>49</b>
	<b>REFERÊNCIAS.....</b>	<b>51</b>



## 1 INTRODUÇÃO

Construir uma máquina ou mecanismo autônomo dotado de inteligência têm sido o foco de pesquisadores das diversas áreas das ciências e da engenharia. Embora os primeiros estudos sobre “Redes Neurais Artificiais” tenham sido publicados há mais de 50 anos, esse assunto começou a ser profundamente pesquisado no início dos anos 90, e ainda possui um enorme potencial de pesquisa. As aplicações envolvendo sistemas considerados inteligentes abrangem uma ampla gama, incluindo: análise de imagens adquiridas de satélites artificiais; classificação de padrões de fala e escrita; reconhecimento facial com visão computacional; controle de trens de alta velocidade; previsão de ações no mercado financeiro; identificação de anomalias em imagens médicas; identificação automática de perfis de crédito para clientes de instituições financeiras (NUNES et al., 2017).

Além dessas aplicações, as capacidades das redes neurais artificiais possibilitam a solução de outros tipos de problemas oriundos de diversas áreas do conhecimento, como para automatizar o diagnóstico de doenças renais, e talvez expandir esse método para outras doenças. Este estudo foca na solução do problema de segmentação de tecidos funcionais dos rins. Enfatiza-se que os rins são duas glândulas localizadas na parte posterior do abdômen que compõem os sistemas de regulação excretora e osmótica e possuem os néfrons como unidades funcionais (ZEDALIS *et al.*, 2018).

A característica mais atraente das redes neurais artificiais neste caso, e também a fonte de sua reputação como ferramentas poderosas para resolver diversos problemas, é sua alta capacidade de mapear sistemas não lineares, permitindo-lhes aprender os comportamentos e características a partir de dados adquiridos de tais sistemas (WALCZAK, 2019).

Apesar de não haver um consenso sobre a definição de inteligência artificial, o presente estudo considera basicamente que inteligência artificial (IA) consiste em sistemas ou máquinas que buscam performar tarefas complexas em constante evolução a partir dos dados coletados (WANG *et al.*, 2019). A inteligência artificial está no centro de novas tecnologias com o potencial de fornecer cuidados de saúde com boa relação custo-benefício e reduzir o tempo de diagnóstico.

Por sua vez, o *deep learning* (Aprendizado profundo), é um subconjunto da inteligência artificial, que consiste em Redes Neurais Artificiais, com várias camadas. A saída de uma camada é geralmente a entrada para a camada subsequente. Quando o modelo contém muitas camadas em sequência, a rede neural é conhecida como rede neural profunda.

Atualmente, o *deep learning* é amplamente utilizado em uma variedade de aplicações, incluindo visão computacional e processamento de linguagem natural (CHEN; RAN, 2019).

A segmentação semântica de imagens é o processo de classificar cada pixel de uma imagem com sua classe correspondente. As abordagens baseadas em Redes Neurais Convolucionais e suas variantes, é a estratégia popular para resolver tarefas de segmentação de imagens médicas. Neste sentido, modelos convolucionais baseados em arquiteturas codificador-decodificador, como U-Net, U-Net++ têm demonstrado um excelente desempenho para segmentação de imagens médicas (JHA *et al.*, 2020).

Na esfera do mercado, hoje diversas empresas e consórcios buscam soluções inteligentes para resolver problemas que antes pareciam não ter solução, como é o caso da Kaggle, fundada em 2010 por Anthony Goldbloom e Jeremy Howard. Seu propósito é intermediar a interação entre governos e empresas com uma comunidade de cientistas de dados, promovendo competições com dados reais para problemas reais. Hoje também oferece uma plataforma de dados pública, uma bancada de trabalho baseada em nuvem para ciência de dados e cursos em inteligência artificial (KAGGLE, 2022).

Uma dessas competições ocorreu durante o período de novembro de 2020 a maio de 2021, chamada de “*HuBMAP - Hacking the Kidney*”. É uma competição que visa dar o primeiro passo no processo de mapeamento do rim humano em resolução unicelular. Muitas áreas da saúde humana seriam afetadas se a ciência entendesse melhor a atividade celular. O desafio é detectar e segmentar unidades de tecido funcional em diferentes amostras do rim. Uma unidade de tecido funcional é definida como um "bloco tridimensional de células centrado em torno de um capilar, de modo que cada célula desse bloco esteja dentro da distância de difusão de qualquer outra célula no mesmo bloco" (DE BONO *et al.*, 2013).

Esta competição foi patrocinada pelo *HuBMAP Consortium - The Human Biomolecular Atlas Program* (Atlas Biomolecular Humano, tradução do autor). Trata-se de um consórcio constituído por 62 instituições de pesquisa que visa criar a próxima geração de ferramentas de análise molecular e tecnologias computacionais para possibilitar a formação de um atlas da função e relação entre as células que constituem o corpo humano. O referido atlas possui o intuito de trazer um melhor entendimento de como as interações intercelulares afetam a saúde humana. Nessa competição específica, o objetivo é segmentar o glomérulo elemento que compõe o néfron, um tecido funcional do rim (ALLEN J *et al.*, 2021).



Diante do exposto, o presente trabalho de conclusão de curso apresenta a construção de um modelo de inteligência artificial, totalmente convolucional, para solucionar o problema proposto pela competição, segmentar tecidos funcionais do rim. O modelo final consiste em uma variação da arquitetura U-Net que obteve uma *performance* superior à 74% dos modelos submetidos à competição. O tempo total de treino, levando em conta todos os experimentos realizados neste trabalho, foi de aproximadamente 70h. Os subtópicos seguintes desta introdução apresentam um detalhamento de outros trabalhos relacionados ao tema, bem como a exposição dos objetivos geral e específicos da pesquisa.

## 1.1 TRABALHOS RELACIONADOS

O estudo de Sarder; Ginley; Tomaszewski (2016) demonstrou a possibilidade de identificar diversas patologias renais de maneira quantitativa e automatizada. O método computacional descrito no estudo suscita que diversos tecidos renais podem ser classificados e analisados quantitativamente de maneira similar a um patologista treinado. As técnicas descritas oferecem um potencial de segmentar características glomerulares importantes, porém exige que o glomérulo tenha sido previamente localizado na imagem de microscopia.

Através de métodos estatísticos e técnicas tradicionais de análise de imagens, Ginley et al. (2017) desenvolveram um algoritmo computacional para identificar, de maneira não supervisionada, glomérulos em imagens de microscópio. O algoritmo foi capaz de localizar 88% dos 1000 glomérulos presentes na imagem de teste. Apesar de promissor, a técnica é demorada, levando entre 6 e 10 segundos para analisar uma imagem que contenha entre 5 e 15 glomérulos.

Por outro lado, a pesquisa de Govind et al. (2018) combinou duas modalidades de microscopia, a imunofluorescência e a coloração PAS para segmentar glomérulos. Combinando as duas imagens e utilizando filtros passa faixa de Butterworth, o algoritmo proposto foi capaz de detectar múltiplos glomérulos em uma única amostra. A *performance* alcançada por este método foi de 83% na métrica DICE, uma métrica comumente utilizada para avaliar sistemas de segmentação, sendo que a nota máxima é de 100%.

Por fim, vale destacar o recente estudo de Ishrak; Khan; Imam (2021) pois utilizaram técnicas de *deep learning* para realizar a segmentação de glomérulos. O trabalho utilizou o banco de dados disponibilizado pelo *HubMap* para treinar uma versão customizada de rede

neural de segmentação LinkNet. O resultado obtido pela rede neural foi de 82,7% na métrica DICE.

A partir dos estudos prévios apresentados, a presente pesquisa busca corroborar para a literatura científica e para a competição do *HubMap* relacionada a segmentação semântica de imagens a partir das Redes Neurais Convolucionais, especificamente na conduta de diagnóstico de doenças relacionadas aos rins.

## 1.2 OBJETIVOS

Os objetivos deste trabalho de conclusão de curso são desenvolver um modelo de inteligência artificial que capaz de segmentar tecidos funcionais do rim, gerando a base tecnológica necessária para auxiliar trabalhos futuros de análise em larga escala dos glomérulos presentes no rim, e assim mostrar o potencial desta tecnologia para o apoio diagnóstico.

### 1.2.1 Objetivo Geral

Treinar um modelo de segmentação semântica de imagens médicas, em várias etapas estruturadas, buscando a melhor *performance* e a capacidade de segmentação de imagens médicas.

### 1.2.2 Objetivos Específicos

Os objetivos específicos do presente estudo são:

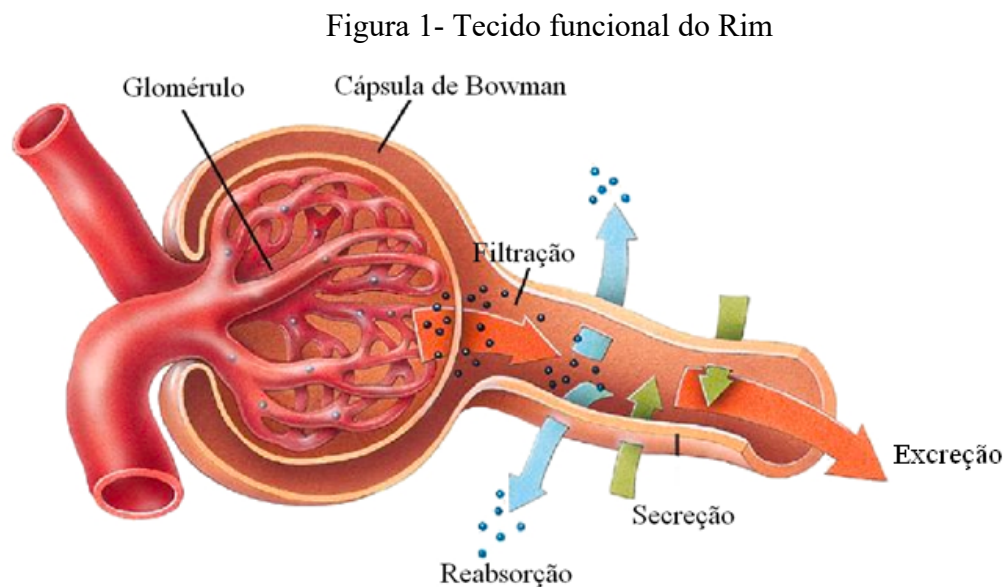
- a) Treinar uma rede neural de segmentação de imagens médicas, capaz de identificar tecidos funcionais do rim, a fim de ajudar patologistas no processo de diagnóstico;
- b) Gerar uma submissão para a competição “*HuBMAP - Hacking the Kidney*”.

## 2 FUNDAMENTAÇÃO TEORICA

Nesta seção serão introduzidos conceitos relevantes para o entendimento do tema do trabalho, como os tecidos funcionais dos rins, as redes neurais artificiais, seus desdobramentos e as redes de segmentação.

### 2.1 TECIDOS FUNCIONAIS DO RIM

Os rins podem ser divididos em três grandes estruturas, o córtex a medula e os cálices. Os néfrons são os principais tecidos funcionais do rim, há cerca de um milhão em cada rim, e são responsáveis pela remoção de resíduos metabólicos do sangue. Um dos principais componentes do néfron é o glomérulo, uma rede de capilares formados por ramos da artéria renal. O sangue passa pelo sistema capilar glomerular e é filtrado, formando assim a urina. Isto é feito através de uma barreira de filtração que filtra seletivamente a água e os solutos do sangue que passam pelos capilares glomerulares (VIEIRA, 2022).



Fonte: <https://www.biomedicinapadrao.com.br/2013/12/entendendo-o-que-e-filtracao-reabsorcao.html>

Além do processo de filtração, os rins apresentam as etapas de reabsorção, secreção e excreção, como ilustradas pela Figura 1. Durante o processo de reabsorção, os aminoácidos, glicose, ureia e água são recapturados pela corrente sanguínea. Já no processo de secreção acontece a remoção de íons de potássio, amônia e hidrogênio (CÂMARA, 2013).

Deste modo, as doenças que danificam o glomérulo permitem que glóbulos vermelhos e brancos, além de proteínas, passem pela barreira de filtração glomerular. Isto ocasiona uma condição em que existe a presença excessiva de proteínas na urina, e pode ser interpretada como um indicativo de diversas doenças renais, além de incidência de danos estruturais no glomérulo. Durante biopsias renais, os especialistas focam na histologia do glomérulo para distinguir os tipos de doenças renais. O ponto é que este processo pode ser automatizado, reduzindo o tempo e o custo do diagnóstico (SARDER; GINLEY; TOMASZEWSKI, 2016). Para que este processo possa ser automatizado, é necessário que primeiro o glomérulo tenha sido localizado na imagem de microscópio. Neste trabalho de conclusão de curso será demonstrado um modelo que mitiga este desafio.

## 2.2 REDES NEURAIS ARTIFICIAIS

As redes neurais artificiais (RNA), inspiradas nas redes neurais biológicas, têm o início do seu estudo datado na década de 40, com a pesquisa de McCulloch e Pitt (1943). Os autores desenvolveram um dos primeiros circuitos elétricos que visava simular um neurônio. Duas décadas depois, os pesquisadores Widrow e Hoff (1962), da Universidade de Stanford, desenvolveram um método de aprendizado para a modelagem matemática do neurônio chamada de *perceptron*.

O *perceptron* realiza a soma ponderada de todas as entradas, acrescidas de um termo de viés ou *bias*  $b$ , onde cada entrada  $X_i$  está relacionada a um peso  $W_i$ . O resultado desta operação é utilizado como argumento para uma função não linear, também conhecida como função de ativação (RODRIGUES, 2018). Observa-se a fórmula matemática que descreve o *perceptron* na equação 1.

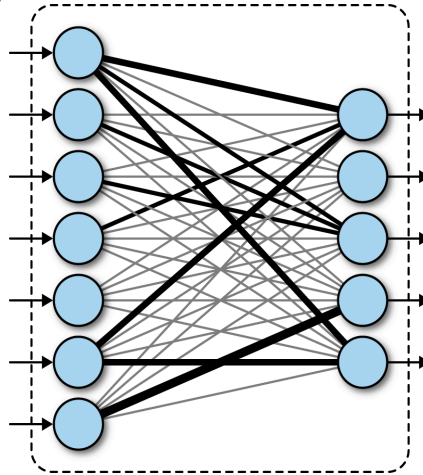
$$f(X) = fa\left(\sum_{i=0}^n X_i * W_i + b\right) \quad (1)$$

As RNA aprendem a realizar tarefas usando algoritmos de aprendizado que geralmente requerem grandes conjuntos de dados. O treinamento consiste em ajustar os pesos  $W$  de cada neurônio presente na rede, a fim de reduzir o erro observado ao comparar a saída da rede com o gabarito (CUNHA, 2020).

### 2.2.1 Redes totalmente conectadas

As redes totalmente conectadas (em inglês: *Fully Connected Networks*) são redes neurais compostas por uma ou mais camadas de *perceptrons* totalmente conectados entre si. Estas camadas totalmente conectadas, são chamadas assim devido ao fato de todas as dimensões de saída serem dependentes de todas as dimensões de entrada, um exemplo ilustrativo pode ser encontrado na Figura 2 (RAMSUNDAR; ZADEH, 2018).

Figura 2 - Redes Totalmente Conectadas



Fonte: <https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>

O principal ponto vantajoso das redes totalmente conectadas é que elas são independentes de estrutura, isto é, não há suposições especiais necessárias a serem feitas sobre o formato da entrada. No entanto, em algumas aplicações com o processamento de imagens, a estrutura do dado de entrada faz parte da própria informação. A imagem é um tipo estruturado de dado, que é essencialmente uma matriz de pixels, onde a localidade de cada pixel também é uma informação relevante (CUNHA, 2020).

### 2.2.2 Redes Neurais Convolucionais

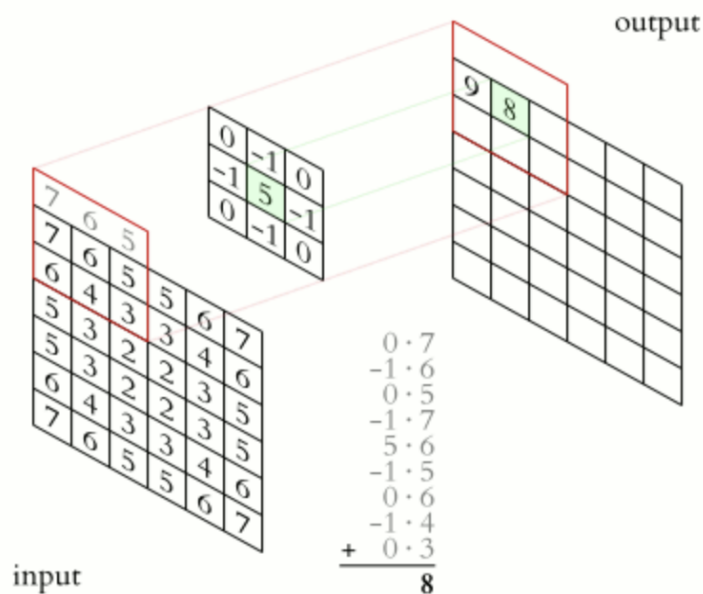
As Redes Neurais Convolucionais (em inglês: *Convolutional Neural Networks*) têm sido o método de escolha para reconhecimento de caracteres escritos a mão desde LeCun et al. (1998). Estas são um caso particular das redes neurais totalmente conectadas, onde os pesos dos *perceptrons* são compartilhados entre diferentes entradas. Na visão de Alake (2020) apenas

recentemente, devido ao aumento do poder computacional e capacidade de armazenamento de dados, as Redes Neurais Convolucionais (CNN) ganharam viabilidade e popularidade na área de processamento de imagens de alta complexidade, incluindo segmentação automatizada de imagens médicas.

Neste âmbito, a convolução se refere à combinação matemática de duas funções para produzir uma terceira. Quando referente a uma CNN, a convolução é realizada nos dados de entrada através de um filtro ou *kernel* para produzir um mapa de ativação ou saída de dados (GIRMA, 2019).

Conforme suscitado por Cunha (2020), esse tipo de rede possui uma estrutura de camadas, onde cada *kernel* de convolução gera um novo mapa, ou imagem, para a próxima camada. Em suma, eles são responsáveis pela identificação de características da imagem. Por fim, os *kernels* de convolução percorrerem a imagem, cada uma em busca de uma característica específica.

Figura 3 - Representação de mapa de ativação através da entrada



Fonte: Disponível em <https://abenezer-g.medium.com/>

Na Figura 3, pode ser observado como é construído o mapa de ativação (*output*) através da entrada (*input*). O filtro é deslizado pela entrada, e a saída corresponde a soma da multiplicação ponto a ponto dos parâmetros do filtro com os parâmetros correspondentes da entrada.

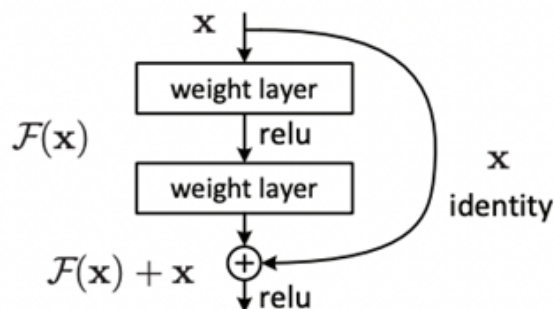
Para Malach e Shalev-Shwartz (2020), a superioridade das redes convolucionais em relação às redes totalmente conectadas pode ser explicada, principalmente, pelo fato de o processo da convolução levar em conta a estrutura da informação de entrada, explorando a correlação parcial entre os pixels, ao contrário das redes totalmente conectadas que descartam qualquer informação espacial da entrada. Além disso, melhora a eficiência de memória por precisar armazenar menos parâmetros, aumentando a sua eficiência computacional.

### 2.2.3 Residual Neural Network (ResNet)

De acordo com Simonyan e Zisserman (2014), geralmente redes neurais mais profundas tendem a exibir melhor performance. Contudo, ao se aumentar o número de camadas em uma rede, mais difícil fica treiná-la. Para os autores, em seu estudo, o treinamento de uma rede de 19 camadas precisou ser realizado em partes, adicionando três camadas de cada vez, após a sétima camada. Corrobora nesta ideia, Bohra (2021), de que conforme o algoritmo de treino avança às camadas mais profundas, os gradientes de treino se tornam cada vez menores, o que eventualmente deixa as últimas camadas com os pesos praticamente intactos. Como resultado, a rede nunca converge a um ponto ótimo.

A ResNet, trouxe uma revolução nesse aspecto, pois permitiu com que redes com uma ordem de magnitude superior de camadas fossem treinadas inteiramente do zero, apenas com a inicialização aleatória dos pesos. Conforme explicado por He et al. (2015), o grande diferencial dessa arquitetura foi adicionar uma soma entre a saída e entrada de blocos de convolução. Isso permitiu uma maior estabilidade durante o processo de treino devido às conexões residuais que eliminaram o problema da diminuição dos gradientes nas camadas mais profundas. O bloco básico de construção da ResNet pode ser verificado na Figura 4.

Figura 4 - Representação do bloco básico da ResNet



He et al. (2015, p.2)

Na Figura 4 é possível observar as camadas de convolução representadas por *weight layer*, e a conexão residual, ou função identidade, representada por *identity*. Curiosamente, importante ressaltar que em 2015, esta arquitetura ganhou a competição de classificação de imagens *ImageNet Large Scale Visual Recognition Challenge* (ILSVRC), utilizando uma *ResNet* com 152 camadas.

#### 2.2.4 EfficientNet

Ao contrário da prática convencional que dimensiona arbitrariamente fatores como número de camadas, número de filtros de convolução em cada camada e resolução de entrada, o método de dimensionamento *EfficientNet* dimensiona a largura, a profundidade e a resolução da rede com um conjunto de coeficientes de dimensionamento (TAN; LE, 2019).

A utilização deste mecanismo é eficiente pois, as *EfficientNets* atingem *performance* igual ou superior aos modelos de estado da arte de redes convolucionais visto que utilizam seis vezes menos poder computacional e utilizam oito vezes menos parâmetros. O principal diferencial desta arquitetura, foi o método utilizado para concebê-la. Primeiro foi desenvolvida uma arquitetura inicial pequena e eficiente de forma algorítmica, que então foi expandida.

Assim como a *ResNet*, a *EfficientNet* também utiliza conexões residuais entre blocos de convolução para permitir o treinamento de redes com mais de cem camadas.

### 2.3 REDES DE SEGMENTAÇÃO

As redes convolucionais são tipicamente utilizadas para resolver problemas de classificação, em que a saída da rede apresenta apenas uma dimensão de saída. Redes de segmentação, geralmente, possuem um pixel de saída para cada pixel de entrada, visando segmentar uma imagem em diferentes partes (ZHANG, 2019).

Ressalta-se que as redes de segmentação semântica geralmente podem ser divididas em dois módulos separados: o codificador e o decodificador. O módulo do codificador usa uma combinação de operações de convolução e *pooling* para extrair características e capturar o contexto global dos dados de entrada. O módulo decodificador recupera os detalhes espaciais perdidos durante a codificação e prevê os rótulos dos pixels (POUDEL; LIWICKI; CIPOLLA,



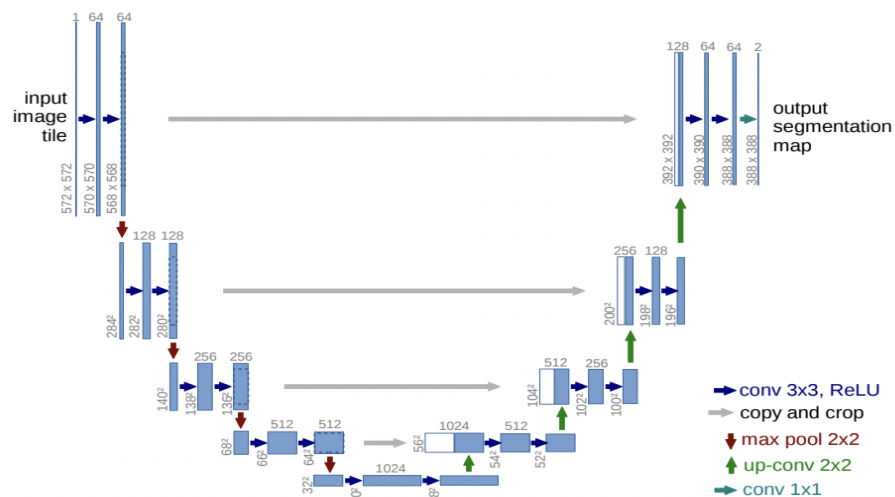
2019). Portanto, esse método possui relevante utilidade no diagnóstico de doenças renais, na qual tecidos de órgãos precisam ser segmentados.

### 2.3.1 U-Net

U-Net é uma arquitetura de segmentação de imagem que consiste em dois blocos principais: O *encoder*, em que a imagem é contraída (*downsampling*) a cada bloco de convolução; e o *decoder*, em que a imagem é expandida. O caminho de contração da imagem não difere da arquitetura de uma rede convolucional que seria usada para uma tarefa de classificação, por exemplo. Por outro lado, o caminho de expansão da imagem envolve uma transformação de *upsampling* que aumenta o tamanho da imagem (RONNEBERGER; FISCHER; BROX, 2015).

Deste modo, enfatiza-se que cada camada do *encoder* produz um mapa de ativação de dimensões equivalentes à sua respectiva camada do *decoder*. Isso possibilita que ambos os mapas sejam concatenados e utilizados com facilidade para produzir a saída da rede.

Figura 5 - Representação da arquitetura da U-Net



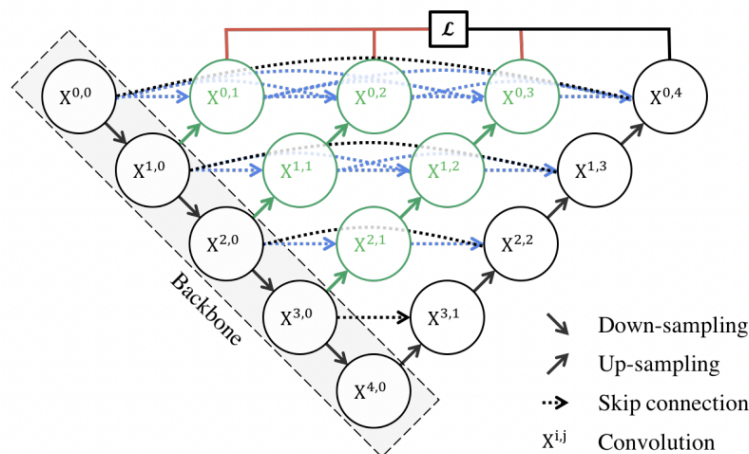
Fonte: Ronneberg; Fischer; Brox (2015, p.02)

Na Figura 5 pode-se observar a arquitetura da U-Net, além de todas as suas operações de convolução. No canto superior esquerdo, apresenta-se o caminho de *downsampling* e, no canto superior direito, apresenta-se o caminho de *upsampling*.

### 2.3.1.1 U-net++

Após a introdução da U-Net, diversas mudanças foram propostas em sua arquitetura original. Uma dessas evoluções foi apresentada por Zhou et al. (2018) e é denominada U-Net++, representada na Figura 6.

Figura 6 - Representação da arquitetura da U-Net++



Fonte: Zhou et al. (2018, p.3)

Conforme exposta na Figura 6, tal arquitetura consiste em um caminho de *downsampling*, também chamado de *encoder*, e um caminho de *upsampling*, também chamado de *decoder*. Conforme explicado por Zhou et al. (2018), o *encoder* e o *decoder* são conectados por uma série de blocos de convolução. A ideia principal destes blocos é reduzir a "lacuna semântica" entre o *encoder* e o *decoder* antes destes serem concatenados e passados para o próximo estágio.

Tabela 1 - Comparação entre U-Net e U-Net++

Architecture	Params	Dataset			
		cell nuclei	colon polyp	liver	lung nodule
U-Net [9]	7.76M	90.77	30.08	76.62	71.47
Wide U-Net	9.13M	90.92	30.14	76.58	73.38
UNet++ w/o DS	9.04M	<b>92.63</b>	<b>33.45</b>	79.70	76.44
UNet++ w/ DS	9.04M	92.52	32.12	<b>82.90</b>	<b>77.21</b>

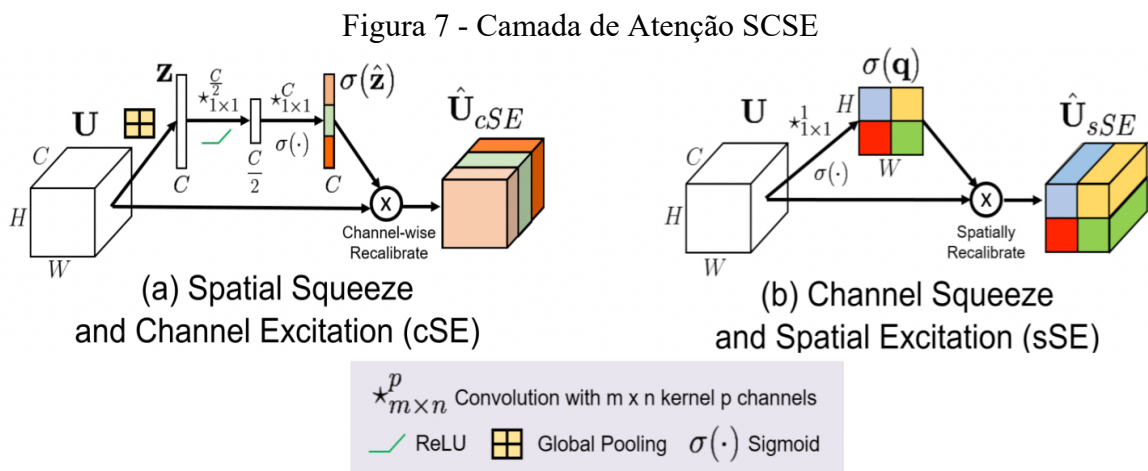
Fonte: Zhou et al. (2018, p.7)

A Tabela 1 confirma a tese apresentada nas pesquisas de Zhou et al. (2018) de que, a U-Net++ se provou superior a U-Net em diversos *datasets* da área biomédica. Neste sentido, os experimentos realizados durante este trabalho de conclusão de curso e que terão seus procedimentos e resultados expostos no próximo capítulo foram realizados utilizando também a U-Net++.

### 2.3.1.2 Camadas de Atenção

Outra modificação proposta na arquitetura de redes totalmente convolucionais voltadas a segmentação foi a adição de camadas de atenção após os blocos de convolução pertencentes aos *encoders* e *decoders*. Neste trabalho considera-se a camada de atenção do tipo SCSE (*Spatial and Channel Squeeze & Excitation*). Essa camada representa a fusão de dois modelos de atenção, sSE (*Spacial Squeeze and Excitation*) ou compressão espacial e excitação e cSE (*Chanel Squeeze and Excitation*) ou Compressão de canais e excitação.

A SCSE busca recalibrar as ativações produzidas pelas camadas de convolução de forma adaptativa, enfatizando características relevantes para a resolução do problema enquanto suprimindo características que sejam irrelevantes (ROY; NAVAB; WACHINGER, 2018).



Fonte: Adaptada de (ROY; NAVAB; WACHINGER, 2018)

Como explicado por Ahmed (2020, p. 8) camadas de atenção do tipo cSE são aplicadas após uma operação de convolução e buscam enfatizar canais de ativação que contenham características relevantes e suprimir os outros canais. Este tipo de camada gera um vetor com um valor escalar para cada canal de ativação. Isto é feito através de uma operação de *pooling*

que gera o valor médio de cada canal e duas operações de convolução, no final da operação, os canais de ativação são multiplicados por este valor escalar, como podemos ver na Figura 7 a.

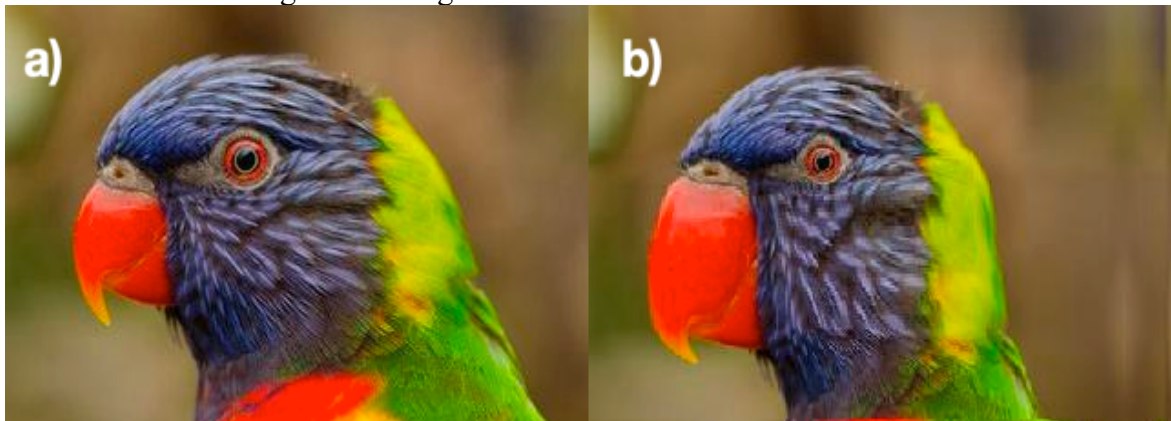
Já camadas de atenção do tipo sSE concatenam os canais de ativação em um único canal, através de uma operação de convolução, gerando uma matriz de pesos que indica a região da imagem que a rede deve focar mais para realizar a segmentação.

## 2.4 DATA AUGMENTATION

Como descrito por Buslaev et al. (2020) *data augmentation* “é um conjunto de técnicas para aumentar a quantidade e diversidade de dados rotulados para treino, através de transformações nos dados de entrada que preservem os rótulos de saída”.

Além de aumentar a quantidade dos dados de treino, esta técnica é um importante meio de regularização para impedir o *overfitting*, tornando a rede mais robusta sem que seja necessário adicionar novos parâmetros treináveis (RODRIGUES, 2018).

Figura 8 - Imagem transformada com *GridDistortion*



Fonte: (APP · STREAMLIT, [s. d.]

Neste trabalho foi utilizado a ferramenta desenvolvida por Buslaev et al. (2020), a *Albumentations*, que possui diversas transformações de imagens para problemas de segmentação e classificação de imagens. Na Figura 8 é representado um exemplo de transformação disponível na ferramenta, chamada de *GridDistortion*, onde o exemplo a) contém a imagem original e o exemplo b) a imagem transformada.

## 2.5 MÉTRICAS DE AVALIAÇÃO E FUNÇÃO DE PERDA

As métricas de avaliação são um componente importante do treinamento de redes neurais, com elas é possível comparar modelos e aferir se estamos fazendo progresso. (BAJAJ, 2022)

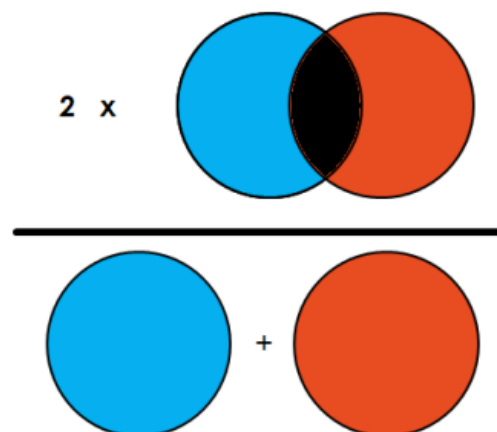
### 2.5.1 Coeficiente DICE

O Coeficiente de similaridade DICE, ou índice de Sørensen–Dice, é uma ferramenta matemática para medir a similaridade entre dois conjuntos de dados. Este coeficiente se tornou um dos mais utilizados para validar máscaras de segmentação produzidas por algoritmos de inteligência artificial (BELL; MOORE, 2020).

Esta ferramenta é utilizada na área de segmentação de imagens para comparar, *pixel a pixel*, a segmentação produzida com um modelo com o gabarito produzido por um especialista. (YERRAM, 2020).

O valor do DICE varia de 0 a 1, sendo 0 totalmente diferente do gabarito e 1 igual ao gabarito. Pode ser calculado por 2 vezes a interseção do número de pixel entre a predição e o gabarito, dividido pela soma da quantidade total de pixels nas duas imagens, como ilustrado pela Figura 9.

Figura 9 - Diagrama de Venn do Coeficiente DICE



A Equação 2 descreve a fórmula matemática da métrica DICE, sendo  $|X|$  o número de pixels da imagem de predição,  $|Y|$  o número de pixels do gabarito e  $|X \cap Y|$  o número de pixels iguais entre X e Y.

$$DICE = \frac{2 * |X \cap Y|}{|X| + |Y|} \quad (2)$$

### 2.5.2 Função de perda

Para qualquer problema de aprendizagem supervisionada, o processo de treino busca selecionar pesos que forneçam a estimativa ótima da função que modela nossos dados. Podemos tratar o problema de treinar uma rede neural como um problema de minimização, onde procuramos uma matriz de pesos  $W$  que minimize a função de perda, ou erro do modelo (GOODFELLOW; YOSHUA; COURVILLE, 2016, p. 200). Deste modo, a função de perda escolhida por esse trabalho é a *Cross Entropy Loss*, a mesma utilizada pelos idealizadores da Unet Ronneberger, Fischer e Brox (2015) em sua pesquisa.

$$FP(Y, X) = - \sum_{c=1}^c Y_c * \log(X_c) \quad (3)$$

Observa-se a fórmula matemática que descreve a função de perda na Equação 3. Onde a previsão realizada pelo modelo X, é avaliada perante o gabarito Y, para cada classe C. Este cálculo é feito pixel a pixel no caso da segmentação semântica.

### 3 DESENVOLVIMENTO

Neste capítulo serão expostas as restrições computacionais que influenciaram na realização dos testes presentes neste estudo, os *datasets* utilizados, e as arquiteturas propostas, incluindo U-Net, U-Net++. Os otimizadores testados e as transformações realizadas no conjunto de treino serão discutidas também.

#### 3.1 RESTRIÇÕES COMPUTACIONAIS

A principal limitação durante a execução dos experimentos realizados nesta pesquisa foi o custo computacional. Tal limitador comprometeu o número de experimentos e tempo de máquina dedicado a cada experimento. Por consequência foi decidido rodar dois tipos de experimentos, curtos e longos. O tamanho do experimento foi limitado pelo número de épocas. Uma época significa uma passagem do modelo por todas as imagens de treino.

Experimentos curtos foram aqueles aos quais foram alocadas cinco épocas de treino para realizá-los. Eles têm por objetivo indicar rapidamente se o experimento realizado apresenta uma melhora incremental em relação ao benchmark analisado. Por outro lado, experimentos longos foram realizados por doze épocas, estes têm como objetivo treinar o modelo sendo avaliado até que este chegue em um platô, uma região na qual o modelo não apresenta melhora significativa mesmo após 2 épocas de treino, o qual neste trabalho foi considerado então ser o limite máximo de *performance* do modelo.

Todos os experimentos foram executados em um servidor disponibilizado pela plataforma Kaggle com uma CPU de 2 núcleos, 13 GB de Memória RAM e 1 Acelerador NVIDIA TESLA P100 com 16 GB de VRAM.

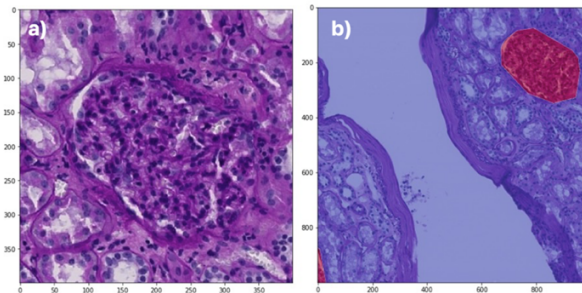
#### 3.2 CONJUNTO DE DADOS

O conjunto de imagens disponibilizadas pelo *HuBMAP* inclui 20 imagens das quais 11 foram obtidas através de amostras recém congeladas e 9 através de amostras fixadas em formalina e incluídas em parafina. Todas as imagens possuem anotações de máscaras de segmentação identificando os glomérulos nas amostras e máscaras classificando a estrutura anatômica do rim entre córtex e medula. Apenas 15 das 20 máscaras de segmentação foram compartilhadas com os participantes da competição, deixando 5 máscaras reservadas para a

avaliação dos modelos submetidos a competição (KAGGLE, 2020). Ademais, dentre as amostras disponíveis para os participantes, foram escolhidas 10 imagens para treino e 5 imagens os testes de validação internos realizados neste trabalho.

Segundo Nyengaard e Kannan (apud KAGGLE, 2020) existem cerca de 600,000 glomérulos em cada rim humano e um glomérulo tipicamente apresenta dimensões entre 100 $\mu$ m a 350 $\mu$ m de diâmetro e um tamanho aproximadamente esférico. Na Figura 10 é apresentada uma imagem de um dos glomérulos a serem segmentados neste dataset.

Figura 10 - a) Glomérulo centralizado, b) Glomérulo destacado em vermelho

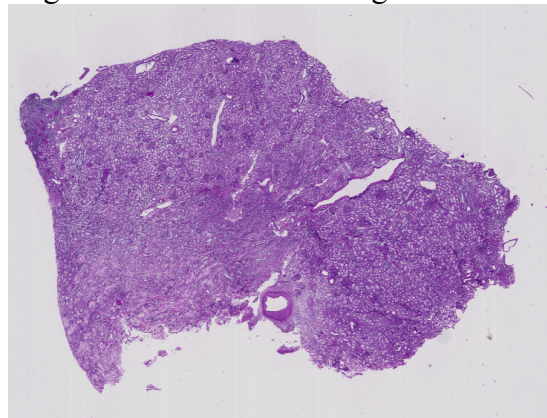


Fonte: Autor

Conforme exposto na Figura 10, o exemplo (a) contém um glomérulo, sem marcação alguma, no centro da imagem. Já na imagem (b), observa-se outro glomérulo segmentado em vermelho, no canto superior da imagem.

Além disso, destaca-se que o *dataset* utilizado é composto de imagens no formato TIFF com tamanhos que variam entre 0,5 Gigabytes e 5 Gigabytes. Uma das amostras utilizadas no *dataset* treino pode ser conferida na Figura 11, esta imagem possui 17.000 por 12.000 pixels.

Figura 11 - Amostra de imagem de treino



Fonte: (ALLEN J *et al.*, 2021)



Para que as imagens fossem processadas com facilidade pelos algoritmos descritos nesse trabalho, foram feitas algumas etapas de pré-processamento, além de extrair recortes menores da imagem original a fim de diminuir o uso de memória durante o seu processamento. Os processos são destrinchados nos tópicos seguintes.

É importante ressaltar que os resultados obtidos neste trabalho são inteiramente baseados nos dados gerados pelo programa HuBMAP: <https://hubmapconsortium.org>, respeitando a política de uso de dados estabelecido pelo consórcio.

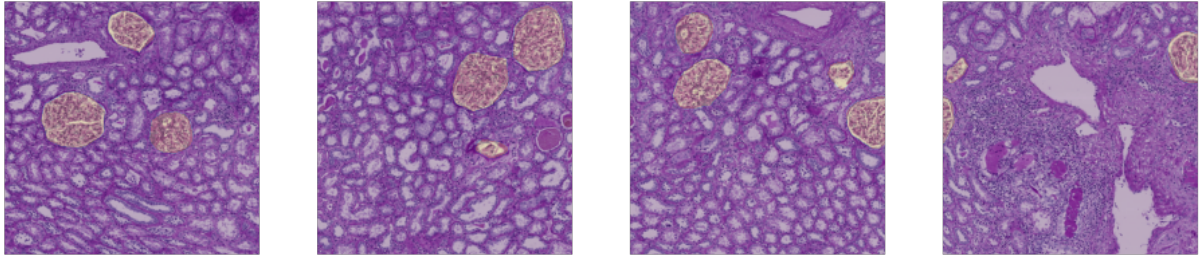
### 3.2.1 Pré processamento

Para facilitar o processo de treino e avaliação, cada amostra de tecido foi reescalada por um fator de  $\frac{1}{3}$ , reduzindo o número de pixels da imagem original e em seguida foram extraídos recortes de 512x512, 384x384, 256x256 pixels. Estes recortes ou *patches* foram salvos no formato .PNG, e agrupados com auxílio das máscaras de classificação anatômica, em três subcategorias: Córtex, Medula e *Background*. O intuito foi reduzir o seu tamanho em disco e facilitar o carregamento nos treinos realizados posteriormente. O número total de recortes obtidos, na resolução de 512x512 pixels, foi de aproximadamente 13.000 imagens. Esta quantidade é 4x maior para os recortes feitos em 256x256 pixels, totalizando 52.000 imagens.

Como já mencionado, o rim apresenta duas áreas principais, cada uma com uma densidade de glomérulos e estrutura diferente, são estas o córtex e a medula. Grande parte dos glomérulos a serem segmentados se encontram no córtex, fazendo com que esta região do rim apresente a grande maioria dos exemplos a serem segmentados. A fim de reduzir a demanda computacional de treinamento foi feita uma amostragem dos *patches* durante o treinamento, fazendo com que *patches* do córtex fossem amostrados com maior frequência. Esse processo foi realizado através de um processo de amostragem com os pesos de 0,7, 0,3 e 0,1 para as classes de córtex, medula e background. Ressalta-se que para o *dataset* de validação, uma amostragem igualitária foi utilizada retirando amostras das 3 regiões em proporções iguais aos presentes na imagem original.

O *split* utilizado para treino consiste em 15 imagens de microscópio, sendo 5 imagens utilizadas para validação e 10 imagens para treino. Na Figura 12 pode ser observado os *patches* das 3 regiões presentes no *dataset*.

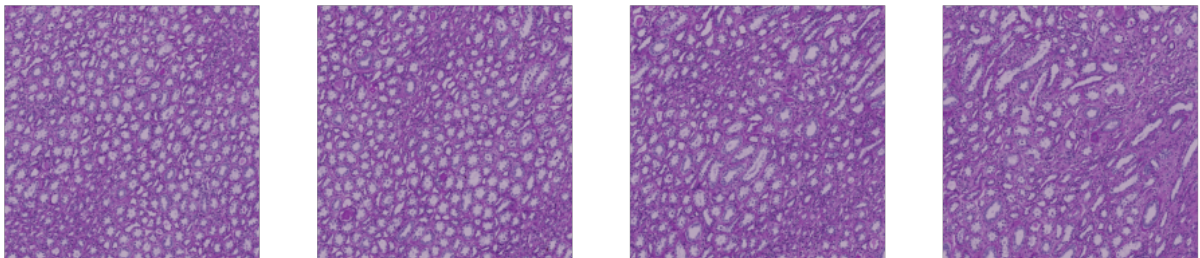
Figura 12 - Recortes do Córtex, glomérulos destacados em amarelo



Fonte: Autor

A Figura 12 representa quatro recortes retirados da região do córtex. Neste caso, os glomérulos podem ser facilmente identificados devido a marcação, feita por especialistas, disponibilizada pelo *dataset* e realizada em um tom ligeiramente amarelado. Podemos ver que em alguns casos, os glomérulos ficam cortados, nas bordas da imagem, tais casos não receberam nenhum processamento especial durante este estudo.

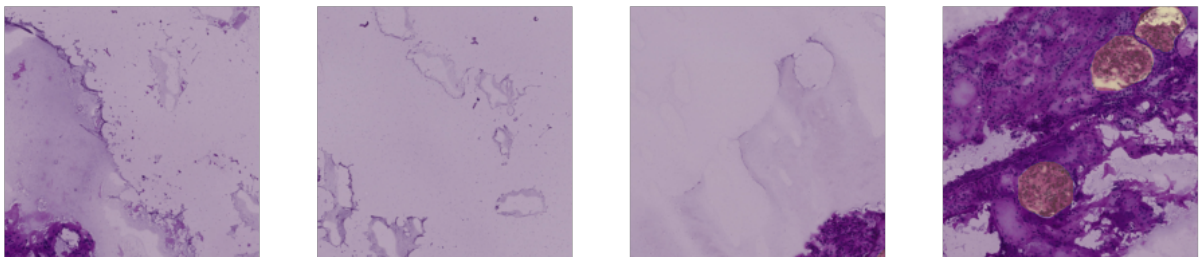
Figura 13 - Recortes da medula



Fonte: Autor

Na Figura 13 o experimento oferece um exemplo de recortes extraídos da região da medula. Tal região apresenta uma menor densidade de glomérulos em relação ao córtex, nos casos apresentados acima, não temos a presença de nenhum glomérulo.

Figura 14 - Recortes do *background*



Fonte: Autor

Já na Figura 14 , os *patches* de *background* são constituídos em sua maioria por imagens que não apresentam tecido a ser avaliado, porém, podem conter glomérulos a serem segmentados como pode ser observado no último recorte a direita.

### 3.3 MODELOS E ARQUITETURAS

Nesta seção encontra-se a apresentação dos modelos utilizados para a realização dos experimentos. O modelo base estudado foi a U-Net, bem como suas variações U-Net++ e modelos de atenção.

#### 3.3.1 Modelo baseline U-Net

Para o modelo baseline foi escolhido uma U-Net, com a implementação apresentada na biblioteca *Segmentation Models* (YAKUBOVSKIY, 2020). A fim de realizar o maior número de experimentos possível, o *backbone* utilizado para a U-Net foi o *Efficient Net B0*, devido a sua *performance* e relativo baixo custo computacional. Em relação a redução do tempo de treino necessário para que o modelo convergisse, os pesos do backbone foram inicializados com os pesos de uma rede equivalente treinada no *dataset* da *ImageNet*. Todos os experimentos conduzidos com essa arquitetura foram do tipo curto, com apenas 5 épocas de treino.

#### 3.3.2 Variações da U-Net

Algumas variações da U-Net *baseline* foram aplicadas, no intuito de obter o melhor *encoder* para a execução da tarefa. A escolha do *encoder* foi baseada na popularidade dos modelos, *performance* apresentada em tarefas de classificação, número de parâmetros, consumo de memória e capacidade computacional exigida. Todos os modelos escolhidos tiveram os seus pesos inicializados a partir de uma rede pré-treinada no *dataset* da *ImageNet*. Os parâmetros dos *encoders* podem ser visualizados na Tabela 2.

Tabela 2 - Arquitetura de *Encoders*

<i>Encoder</i>	Número de Parâmetros (Milhões)
----------------	--------------------------------

---

<i>resnet18</i>	11
<i>resnet34</i>	21
<i>resnet50</i>	23
<i>resnet101</i>	42
<i>efficientnet-b0</i>	4
<i>efficientnet-b1</i>	6
<i>efficientnet-b2</i>	7
<i>efficientnet-b3</i>	10
<i>efficientnet-b4</i>	17
<i>efficientnet-b5</i>	28
<i>efficientnet-b6</i>	40

Fonte: Autor

Esta etapa também foi avaliada usando o tipo de experimentos curtos, devido ao alto número de modelos sendo testados. Diferentemente de todos os outros experimentos que utilizaram a métrica de validação DICE como parâmetro de sucesso, no experimento avaliando tais arquiteturas, optou-se como métrica de sucesso a capacidade do modelo minimizar a função de perda. Uma rede neural que apresenta uma melhor *performance* na função de perda e uma menor *performance* na métrica de avaliação pode ser considerado como um modelo que apresenta uma tendência ao *over-fitting*, apesar de contraintuitivo, existem diversas técnicas que podem ser utilizadas em tais modelos para melhorar a sua *performance*, como apontado por Zaman Khan et al. (2015) o mesmo não é verdade para modelos que apresentam *under-fitting*.

### 3.3.2.1 Modelos de atenção

Durante a fase final dos testes foi realizada a adição da camada de atenção do tipo SCSE. Tal camada foi adicionada após cada bloco de convolução da U-Net. Este experimento foi realizado no formato de experimento longo, buscando avaliar se tal mudança aumentaria a *performance* máxima atingida pelo modelo. Devido a esta escolha, este experimento foi um dos

últimos a serem realizados, isto fez com que fosse necessário apenas executá-lo em um único modelo, reduzindo o custo de teste.

### 3.3.2.2 *U-Net ++*

Na última etapa, o experimento foi conduzido a avaliação da *U-Net++*, uma variação da arquitetura da *U-Net* padrão, conforme já descrito na fundamentação teórica. Este modelo apresenta um consumo de memória maior e realiza mais operações de convolução interna quando comparado a *U-Net*, isto faz com que esta seja a arquitetura mais custosa testada nesta pesquisa. Devido ao alto custo de treinar tal modelo, este foi o último experimento a ser realizado, aplicando todos os melhores resultados dos experimentos de otimizadores, transformações, e tamanhos de imagens anteriores em conjunto com este modelo.

## 3.4 OTIMIZADORES

Nesta pesquisa foram testados alguns otimizadores, no início do processo de treino. Os hiper parâmetros utilizados em cada treino foram os valores padrões sugeridos pela biblioteca *Pytorch* (PASZKE *et al.*, 2019). Os otimizadores testados foram: *SGD*, *Adam*, *RMSProp*, *Ranger*. Tais experimentos ocorreram no formato de experimento curto, devido ao alto número de testes realizados.

## 3.5 DATA AUGMENTATION

A fim de melhorar a *performance* de teste dos modelos treinados, foram adicionadas diversas transformações nas imagens originais. Estas transformações têm o objetivo de aumentar o número de imagens com propriedades diferentes das que o modelo é exposto durante o treino, fazendo com que ele tenha uma maior chance de aprender a resolver o problema proposto. Dessa forma, utilizou-se as transformações implementadas pela biblioteca *Albumentations*.

Durante a exploração inicial dos dados notou-se uma grande variação de brilho e contraste nas diferentes amostras de tecido. Este fator influenciou na decisão de testar a transformação *Random Brightness Contrast*, a qual, como o nome sugere, altera de forma aleatória o brilho e contraste da imagem sendo utilizada. A escolha de utilizar a *GridDistortion*

foi feita devido a esta ser comumente utilizada durante a análise de imagens biomédicas, alterando o formato dos glomérulos.

O tamanho de um glomérulo pode variar até 2.5x e não possui orientação preferencial aparente (SAMUEL *et al.*, 2007). Devido a estas características, foram incluídas no teste a transformação *Shift Scale Rotate*, que engloba uma transformação de rotação e escala. Além disso, foram aplicadas as ações de modificação de orientação de imagem: *HorizontalFlip*, *VerticalFlip*, *RandomRotate90*.

Foi escolhido treinar o modelo com imagens em partes parcialmente ocultas, levando a escolha da transformação *cutouts*, que oculta pequenas partes da imagem durante o processo de treino. Dessa forma, as transformações foram agrupadas em 5 grupos distintos, buscando testar os aspectos de orientação, escala e formato. Por fim, devido ao alto número de testes necessários para avaliar este experimento, ele foi conduzido no formato curto, e a métrica de sucesso escolhida foi o escore de validação DICE.

### 3.6 TAMANHO DO *PATCH*

Outro processo relevante é que os experimentos foram conduzidos testando também o tamanho das imagens recortadas a partir das imagens originais. Os tamanhos escolhidos foram 256x256, 384x384, 512x512. Neste sentido, uma imagem maior significa que a rede possui uma amostra maior de tecido a ser analisado, assim como um contexto maior para realizar a sua segmentação. Além disso, o uso de diferentes tamanhos de imagem faz com que o número de recortes das imagens originais se altere, tornando os modelos treinados em mais, ou menos exemplos, para uma mesma imagem original.

## 4 EXPERIMENTOS E RESULTADOS

Inicialmente, enfatiza-se que quando não especificados no texto, todos os experimentos realizados foram do tipo curto, com recortes de 256x256 pixels retirados a partir das amostras originais do *dataset* de treino. O melhor resultado obtido em cada experimento foi utilizado como *baseline* para cada experimento subsequente.

### 4.1 OTIMIZADORES

O primeiro experimento realizado teve o objetivo de determinar o otimizador a ser utilizado por todos os experimentos subsequentes. Assim, analisou-se a *performance* atingida pelos modelos treinados com os otimizadores listados na Tabela 3, por meio do score DICE obtidos no *dataset* de validação, assim como resultados qualitativos avaliados através do melhor e pior caso obtido.

Ressalta-se que o modelo escolhido para executar tal experimento foi a U-Net, utilizando como *encoder* uma *Efficient Net-B0* com os pesos pré-treinados no *dataset* da *ImageNet*.

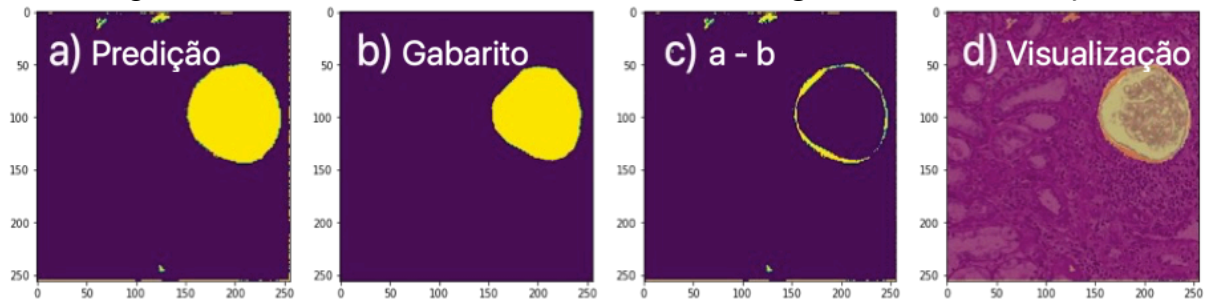
Tabela 3 - Resultado do experimento de otimizadores, *dataset* de validação

<i>Encoder</i>	<b>Otimizador</b>	<b>DICE</b>
<i>efficientnet-b0</i>	<i>ADAM</i>	81,52%
<i>efficientnet-b0</i>	<i>SGD</i>	5,75%
<i>efficientnet-b0</i>	<i>RANGER</i>	84,09%
<i>efficientnet-b0</i>	<i>RMSPROP</i>	85,97%

Fonte: Autor

Observando a Tabela 3 visualiza-se que o melhor otimizador para a tarefa foi o *RMSProp*, seguido pelo otimizador *ranger*. Em seguida foram extraídos os melhores e piores casos de ambos os modelos treinados com tais otimizadores, conforme exibido pela Figura 15, 16, 17, 18.

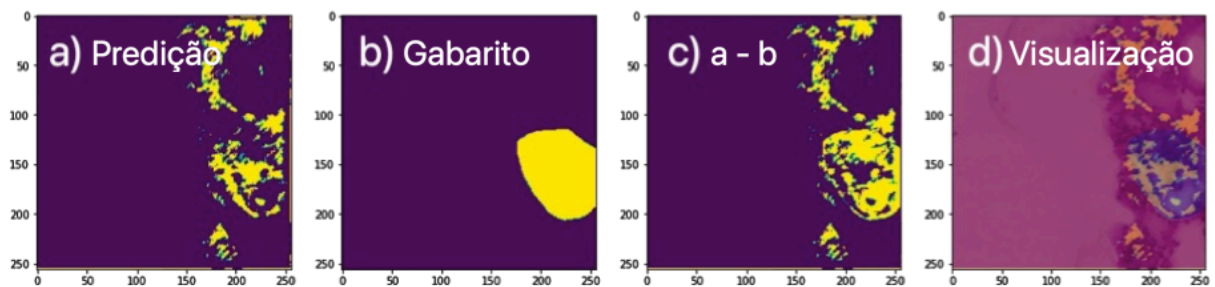
Figura 15 – Melhor caso obtido com o otimizador *ranger*, *dataset* de validação



Fonte: Autor

Na Figura 15 observa-se que o melhor caso obtido com o otimizador *ranger*. A imagem (a) representa a máscara de segmentação produzida pelo modelo, já o exemplo (b) representa o gabarito. Está exemplo foi segmentado satisfatoriamente, apesar da área de segmentação ser um pouco maior que a máscara de treino, conforme podemos ver a imagem (c) a qual indica a diferença entre a máscara gerada pelo modelo (a) e o gabarito ilustrado por (b). Acentua-se que neste exemplo foram os artefatos produzidos por tal modelo, que geraram diversos pixels segmentados perto das bordas da imagem. De outro lado, na Figura 16 expõe-se o pior caso obtido com o otimizador *ranger*.

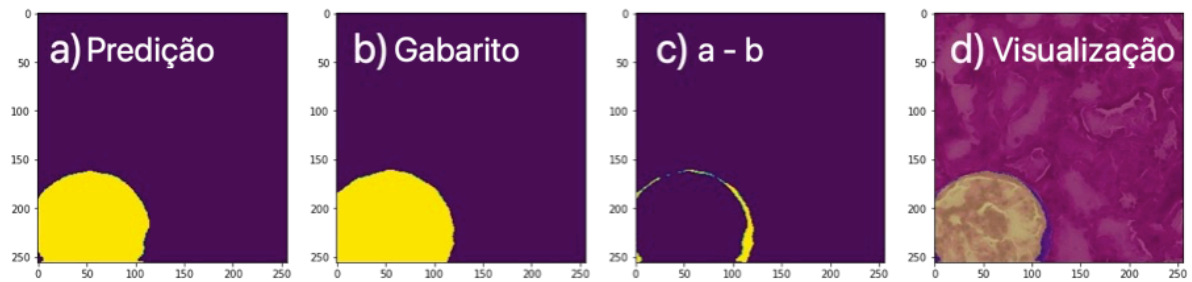
Figura 16 – Pior caso obtido com o otimizador *ranger*, *dataset* de validação



Fonte: Autor

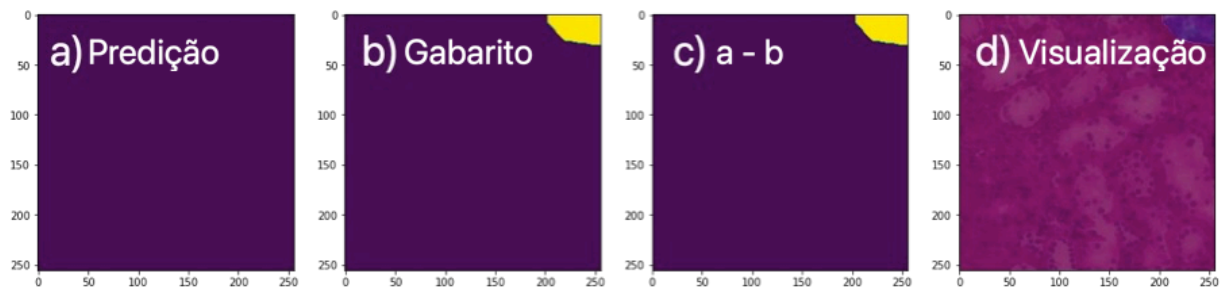
Na Figura 16, portanto, observa-se o pior caso produzido pelo modelo treinado com tal otimizador, especialmente na imagem (a). Os erros de segmentação são visíveis e grosseiros, deixando a maior parte do gabarito não segmentada, como exposto na imagem (c) que representa a diferença entre (a) e (b). Logo, foram testados outros otimizadores, como o RMSProp exibido na Figura 17.



Figura 17 - Melhor caso obtido: RMSProp, *dataset* de validação

Fonte: Autor

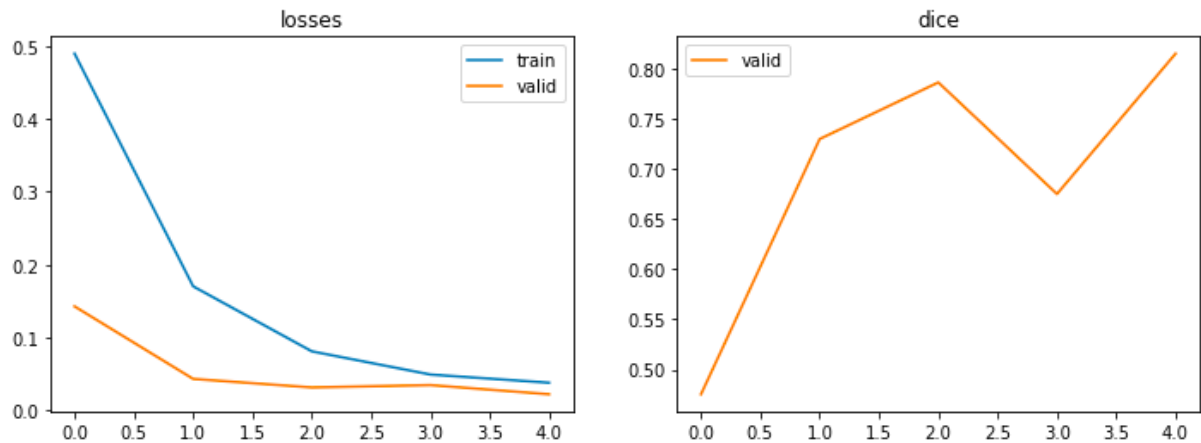
No modelo treinado com o RMSProp, o melhor caso do pode ser visualizado na Figura 17 (a), que também atingiu o melhor escore na métrica de validação. É possível observar uma imagem livre de artefatos, com a sua única imperfeição sendo que a segmentação do glomérulo produzida tem uma área menor do que a segmentação proposta pelo *dataset*, visível no exemplo (d) onde a área do gabarito está marcada em azul e a área segmentada pelo modelo está pintada em amarelo, além do fundo que representa a imagem de entrada. Contudo, na Figura 18 pode ser interpretado o oposto.

Figura 18 - Pior caso obtido: RMSProp, *dataset* de validação

Fonte: Autor

Tal modelo está longe de ser perfeito. Como pode-se inferir na Figura 16, o modelo treinado com o RMSProp apresentou dificuldades em segmentar o glomérulo no canto superior direito da imagem, produzindo uma máscara de segmentação totalmente nula, averiguado na imagem (a). Em seguida, na Figura 19 veja-se as curvas de treino.

Figura 19 - Curvas de treino e validação: RMSProp



Fonte: Autor

Na Figura 19 pontua-se as curvas de treino e validação produzidas pelo modelo treinado com o RMSProp. Devido a sua *performance* superior na métrica de validação, e a ausência de artefatos produzidos nas imagens, este otimizador foi escolhido para realizar todos os experimentos subsequentes.

É importante ressaltar que em nenhum momento foi feita uma busca da melhor taxa de aprendizagem para cada otimizador, uma vez que realizar este experimento seria custoso e exigiria múltiplas passagens de treino para cada otimizador. Ademais, conforme descrito na Tabela 3 que o desempenho do *SGD* foi particularmente afetado por esta decisão, uma vez que este otimizador não possui nenhum mecanismo interno para atualizar a taxa de aprendizado inicial.

#### 4.2 VARIAÇÕES DE ARQUITETURA

Neste segundo experimento buscou-se o modelo que melhor performasse na minimização da função de perda em 5 épocas. A arquitetura escolhida para o experimento foi a da U-Net, com os *backbones* ou *encoders* apresentados na Tabela 4. O peso de todos os *backbones* foi inicializado com os seus respectivos pesos pré-treinados no *dataset* da *ImageNet*.

Tabela 4 - Resultados de diferentes encoders

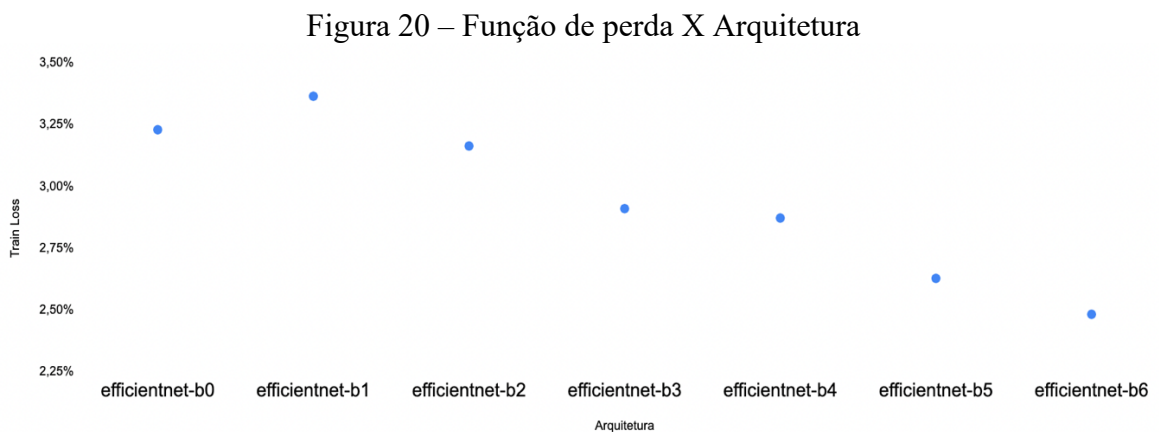
<i>Encoder</i>	<b>Função de perda (dataset de treino)</b>	<b>DICE (dataset de validação)</b>
<i>efficientnet-b0</i>	3,23%	82,70%

<i>efficientnet-b1</i>	3,36%	88,10%
<i>efficientnet-b2</i>	3,16%	88,50%
<i>efficientnet-b3</i>	2,91%	86,02%
<i>efficientnet-b4</i>	2,87%	88,99%
<i>efficientnet-b5</i>	2,63%	83,56%
<i>efficientnet-b6</i>	<b>2,48%</b>	86,55%
<i>resnet18</i>	3,45%	<b>90,26%</b>
<i>resnet 34</i>	3,38%	89,32%
<i>resnet 50</i>	4,24%	87,74%
<i>resnet101</i>	4,35%	89,04%

Fonte: Autor.

Na Tabela 4 observa-se que todos os *encoders* apresentam uma *performance* acima de 80% na métrica de validação DICE. Conforme apontado por Zaman Khan et al. (2015), *overfitting* é nome dado quando uma rede neural durante a fase de treino não melhora a sua capacidade de resolver o problema proposto, porém aprende características aleatórias presentes apenas no *dataset* de treino, piorando a sua *performance* no *dataset* de validação. O contrário, *underfitting* é quando o modelo é incapaz de aprender as características presentes no *dataset* de treino, limitando a *performance* atingida no *dataset* de avaliação.

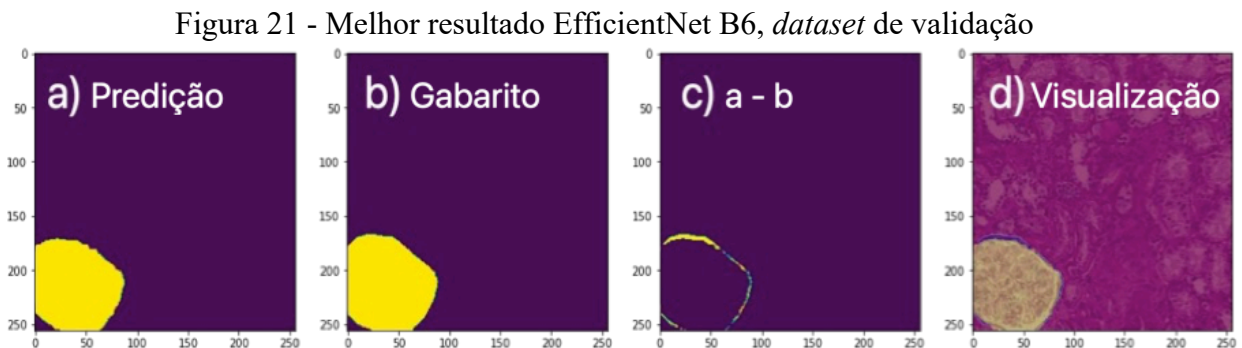
Para a arquitetura da *EfficientNet* é possível notar uma correlação entre o tamanho do modelo e a capacidade de minimizar a função de perda. Isto fica ainda mais evidente no gráfico da Figura 20.



Fonte: Autor

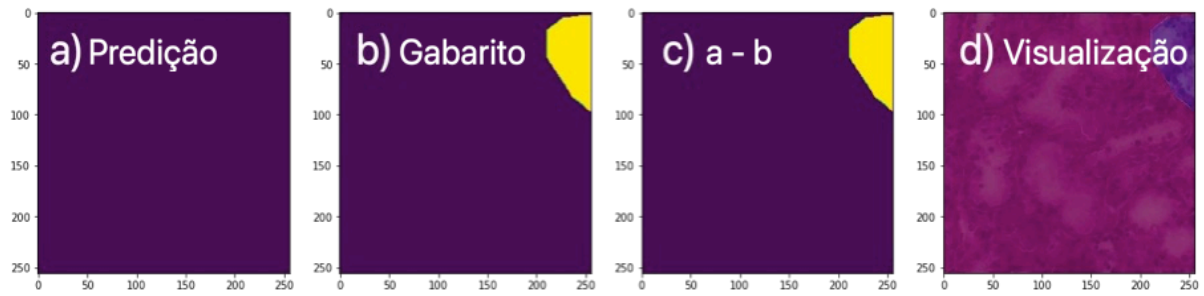
Apesar dos *encoders* do tipo *ResNet* apresentarem uma *performance* mais eficiente no conjunto de validação, eles apresentam uma capacidade menor, em relação aos *encoders* do tipo *EfficientNet*, de aprender as características presentes no *dataset* de treino, indicando uma possível tendência ao *under-fitting*. Por outro lado, os *encoders* do tipo *EfficientNet* apresentam uma possível tendência ao *over-fitting*. Por outro lado, os *encoders* do tipo *EfficientNet* apresentam uma possível tendência ao *over-fitting*.

Como apontado por Zaman Khan et al. (2015), há pouco que possa ser feito para um modelo que apresente *under-fitting*, a melhor alternativa seria treiná-lo por mais épocas. No entanto, isto é geralmente mais custoso do que introduzir variância no conjunto de treino através de técnicas de *data augmentation*, que seriam alternativas para reduzir o *over-fitting*. Por este motivo decidiu-se selecionar o modelo que apresentasse a maior capacidade de minimizar a função de perda no conjunto de treino, e utilizar técnicas de *data augmentation* para reduzir o *over-fitting*. Neste contexto, o *backbone* mais adequado para solucionar o problema proposto foi a *Efficient Net-B6* e ele foi utilizado para conduzir os próximos experimentos, por exemplo, na Figura 21.



Fonte: Autor.

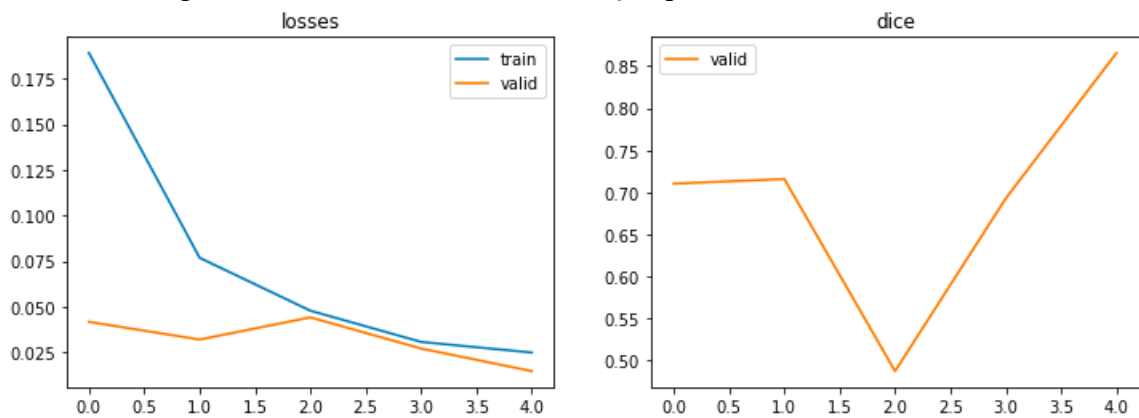
Na Figura 21 há um exemplo de uma segmentação bem-sucedida do modelo escolhido. Nota-se na imagem (d) que o modelo está segmentado o glomérulo em sua forma geral, porém ainda permite pequenas bordas fora da sua segmentação, como pode ser visto no exemplo (c) que representa a diferença entre a segmentação produzida pelo modelo e o gabarito. Contudo o modelo está longe de estar perfeito, conforme a Figura 22.

Figura 22 - Pior resultado EfficientNet B6, *dataset* de validação

Fonte: Autor.

No caso representado na Figura 22, o modelo escolhido não segmentou nada, vide exemplo (a), deixando o glomérulo presente na imagem de treino totalmente de fora. Vale ressaltar que tal comportamento pode ser observado também no primeiro experimento. Esta situação pode ser interpretada como uma dificuldade do modelo em segmentar regiões próximas às bordas das imagens. Avalia-se a seguir na Figura 23, os gráficos das curvas de treino e validação.

Figura 23- Curvas de treino e validação para a Unet-Efficient B6



Fonte: Autor

Na Figura 23 é possível observar uma menor *loss* de validação em relação ao treino. Isso pode ser explicado pela diferença na composição dos *datasets* de treino e validação, onde o *dataset* de treino é artificialmente exposto a mais glomérulos a serem segmentados.

#### 4.3 DATA AUGMENTATION

Este experimento buscou diminuir o *over-fitting* apresentado no experimento anterior. O objetivo é maximizar métrica de validação DICE, fazendo com que a rede apresente uma boa

*performance* no conjunto de validação. Portanto, foram escolhidos grupos de transformação para serem avaliados, conforme discutido na seção 3.5 e compilados na Tabela 5.

Tabela 5 - Resultados do experimento de *Data Augmentation*, *dataset* de validação

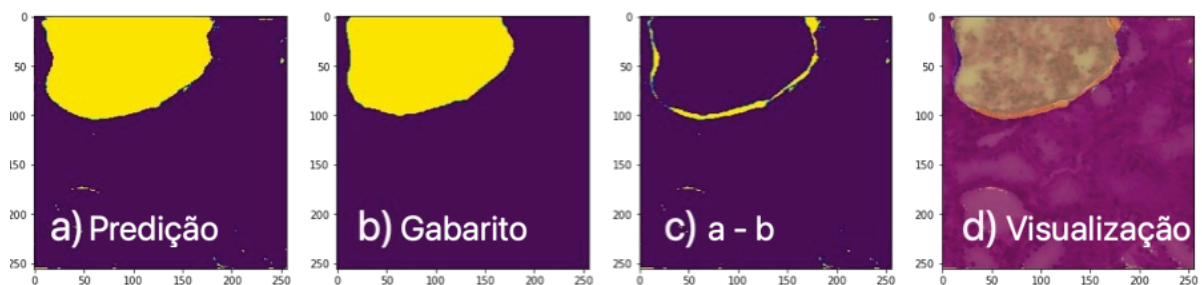
<b>Augmentation</b>	<b>Função de Perda</b>	<b>DICE</b>
<i>Horizontal Flip, GridDistortion</i>	8,36%	87,92%
<i>HorizontalFlip, VerticalFlip, RandomRotate90, ShiftScaleRotate</i>	<b>4,66%</b>	88,25%
<i>HorizontalFlip, GridDistortion, RandomBrightnessContrast, Cutout</i>	9,20%	88,32%
<i>HorizontalFlip, ShiftScaleRotate, GridDistortion, RandomBrightnessContrast</i>	4,88%	89,42%
<i>Horizontal Flip, GridDistortion, RandomBrightnessContrast</i>	5,81%	<b>90,16%</b>

Fonte: Autor.

Analisa-se que na Tabela 5 que apesar da função de perda ter sido maior que o resultado obtido no experimento anterior, todos os tipos de transformação produziram uma métrica de validação melhor em relação ao experimento 2. Isso indica que o modelo treinado a partir de imagens transformadas foi capaz de generalizar melhor o problema proposto, praticamente igualando a *performance* do melhor modelo do experimento anterior.

O melhor resultado foi obtido utilizando "*Horizontal Flip, Grid Distortion e RandomBrightnessContrast*", e este grupo de transformações será utilizado em todos os experimentos a seguintes.

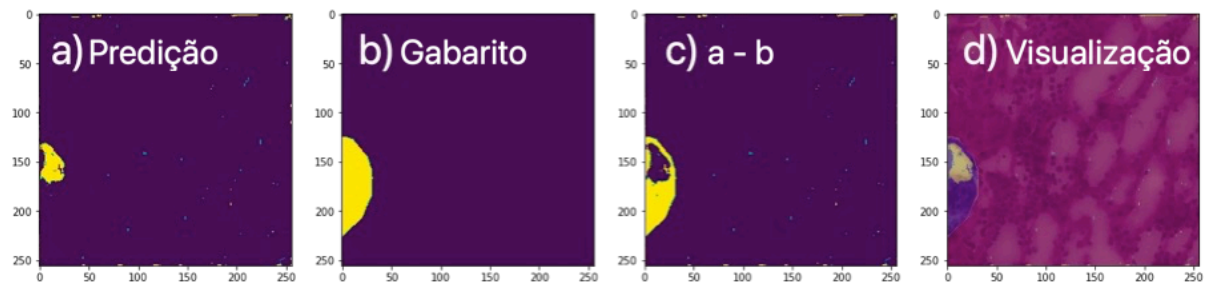
Figura 24 – Melhor resultado com transformação, *dataset* de validação



Fonte: Autor

Na Figura 24 é possível ter uma representação visual do melhor resultado produzido pelo modelo treinado com este grupo de transformações. Averigua-se no exemplo (a) que a predição do modelo captura a forma geral do glomérulo.

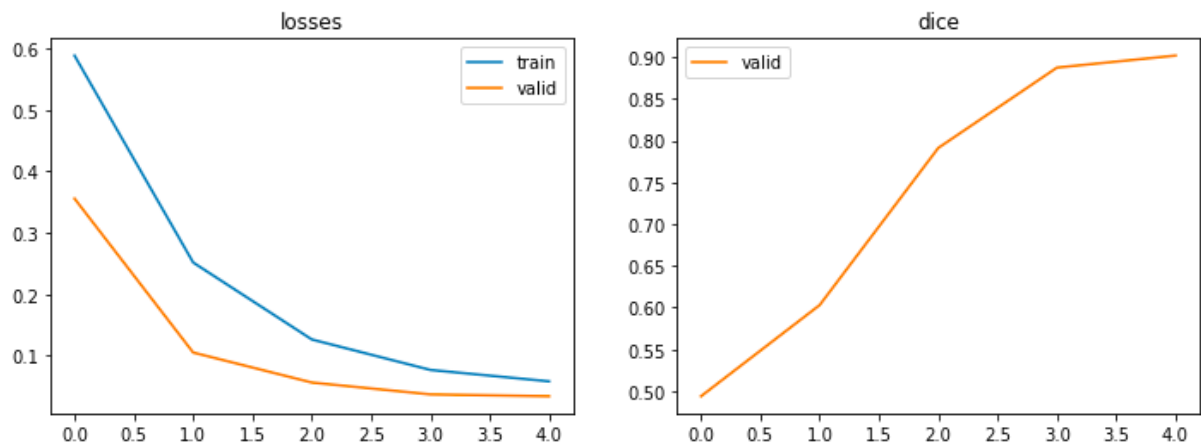
Figura 25 – Pior resultado: Horizontal Flip, Grid Distortion e RandomBrightnessContrast, *dataset* de validação



Fonte: Autor

Em ambos os casos, nota-se uma série de artefatos, pequenos pixels segmentados e dispersos na imagem. Isso não aconteceu nas imagens treinadas sem transformação, apesar da métrica de avaliação ter sido maior. Averigua-se as curvas de treino e validação na Figura 26.

Figura 26 - Curvas de treino e validação do modelo treinado com *Horizontal Flip, Grid Distortion e RandomBrightnessContrast*



Fonte: Autor

Conclui-se uma maior estabilidade na curva de validação DICE em relação ao resultado obtido no experimento anterior, em que a cada época de treino a sua *performance* aumenta.

#### 4.4 TAMANHO DO PATCH E BATCH SIZE

Nesse experimento, devido ao baixo número de casos a ser testado, se tornou viável conduzi-lo no formato longo, treinando os modelos por 12 épocas. Isso permitiu comparar também a diferença de *performance* obtida ao treinar o modelo por 5 e 12 épocas.

Tabela 6 - Resultados para número de épocas diferentes, *dataset* de validação

<b>Resolução</b>	<b>Épocas</b>	<b>Validação DICE</b>
256x256	5	90,16%
256x256	12	<b>91,83%</b>

Fonte: Autor.

Na Tabela 6, ao treinar o modelo obtido pelo experimento anterior por mais 7 épocas, obteve-se um ganho de 1,67% na métrica DICE, indicando que o modelo atinge uma *performance* maior ao ser treinado por mais épocas.

Durante a execução deste experimento foi necessário alterar a quantidade de imagens processadas em um único passo de otimização, também conhecido como *Batch Size*. Isso aconteceu devido ao tamanho da imagem de entrada estar diretamente relacionado ao consumo de memória RAM utilizado para executar o treinamento e avaliação.

Para o teste com imagens de 512x512 pixels foi alterado o tamanho do *batch size* de 16 para 8 imagens, no intuito da memória requerida pelo programa se tornar menor que a memória disponível no acelerador utilizado para treinar o modelo. É apresentado na Tabela 7, o número de imagens em um único *batch size*, onde o consumo de memória e o resultado da métrica de validação DICE é obtido. Como o tamanho do *batch size* influencia diretamente no funcionamento do processo de otimização e treino, tal hiper parâmetro precisa ser levado em consideração junto com o aumento da resolução.

Tabela 7 - Resultados para resoluções diferentes, *dataset* de validação

<b>Batch Size</b>	<b>Resolução</b>	<b>DICE</b>	<b>Consumo de Memória (GB)</b>
16	256x256	91,83%	5,9

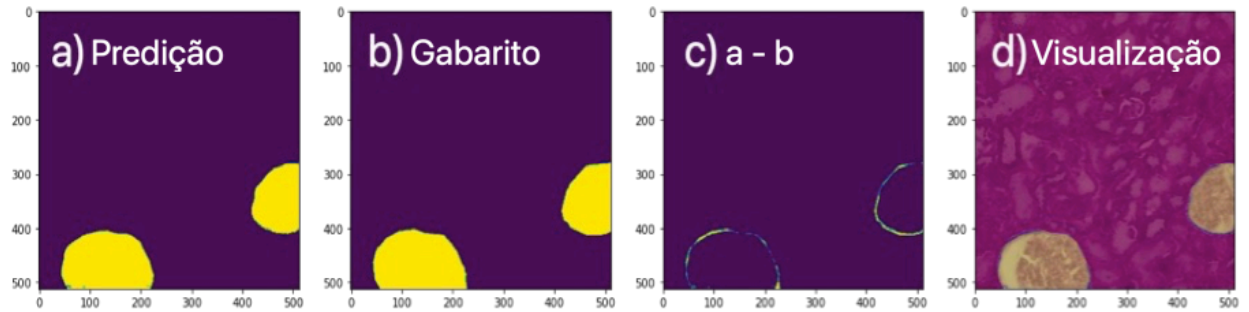


16	384x384	93,04%	13,2
8	512x512	<b>94,09%</b>	12,8

Fonte: Autor

Na Tabela 7, observa-se que o modelo treinado com a maior resolução apresenta o melhor resultado na métrica de validação DICE, além disso, é possível notar uma correlação positiva entre o tamanho da imagem de entrada e a *performance* dos modelos. Veja-se o melhor caso na Figura 27.

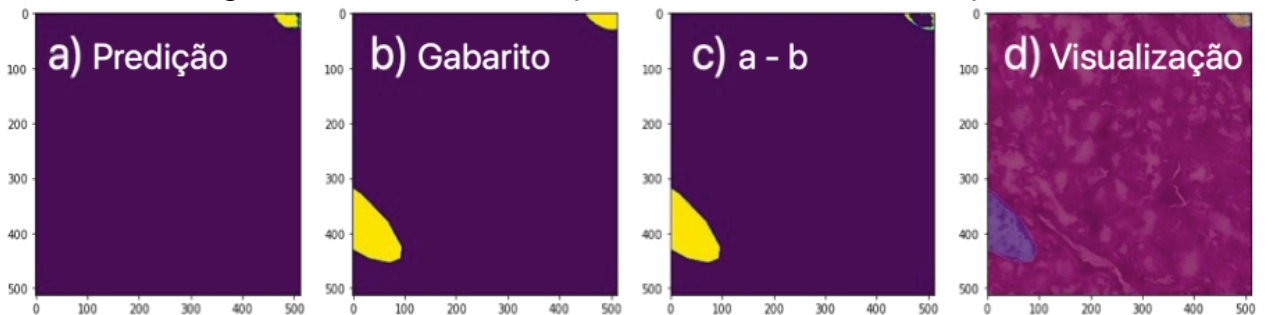
Figura 27 - Melhor caso: resolução 512x512, *dataset* de validação



Fonte: Autor.

Na Figura 27 (a), que representa a máscara de segmentação produzida pelo modelo, conclui-se que com mais épocas de treino e uma maior resolução, os artefatos presentes na seção 4.3 sumiram. O escore atingido neste único recorte foi de 98% na métrica DICE, deixando apenas as bordas do glomérulo sem segmentação, como pode-se averiguar no exemplo (c) que representa a diferença entre o gabarito (b) e o *output* do modelo (a). Apesar do modelo ter feito um ótimo trabalho neste exemplo, ele ainda apresenta falhas, como pode ser visto na Figura 28.

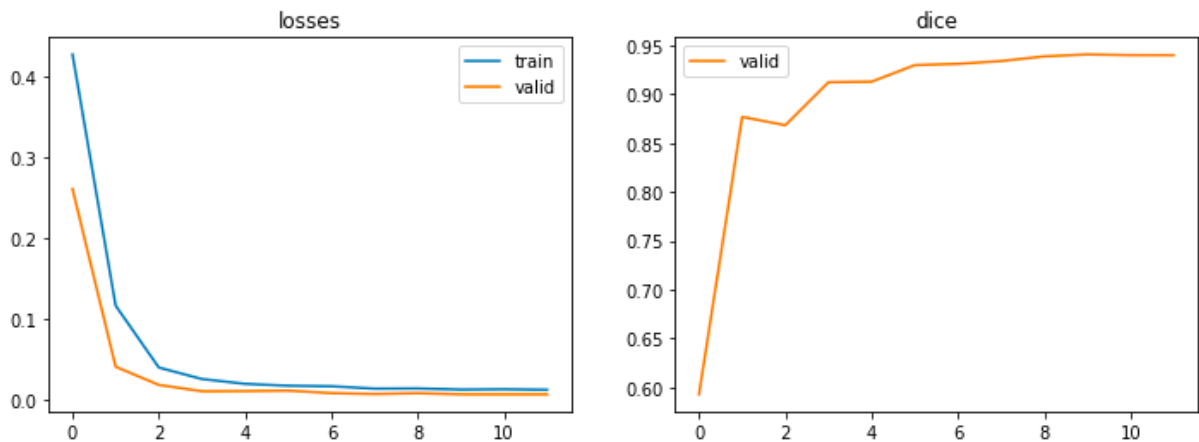
Figura 28 - Pior caso: Resolução 512x512, *dataset* de validação



Fonte: Autor.

Na Figura 28 constata-se que o modelo segmentou parte do glomérulo presente no canto superior direito da imagem, porém falhou em segmentar o outro glomérulo presente no canto inferior esquerdo. Isto confirma que o modelo, apesar de não apresentar os artefatos do item anterior, ainda apresenta problemas, deixando de segmentar regiões importantes da imagem. Tal problema tem sido reportado nos experimentos anteriores. As curvas de treino, por conseguinte, são expostas na Figura 29.

Figura 29 - Curvas de treino e validação para  $batchsize=8$  e resolução= $512 \times 512$



Fonte: Autor

Nos gráficos de treino, assim como no gráfico de validação, da Figura 29, é possível constatar que a *performance* do modelo chega a um platô durante as últimas épocas de treino, não apresentando melhorias consideráveis após a oitava época. O resultado obtido com este experimento melhora a métrica de validação DICE 3,93% em relação ao experimento anterior, um dos maiores incrementos de *performance* registrados pelos experimentos realizados neste trabalho.

#### 4.5 MODELOS DE ATENÇÃO

Neste experimento foi implementado uma variação da U-Net padrão conforme aplicada nos experimentos anteriores. Tal variação consiste em adicionar uma camada de "Atenção" descrita na seção 3.1.1.1. O modelo foi treinado utilizando todos os melhores resultados dos experimentos anteriores e comparado com a *Baseline*, obtida a partir do

experimento da seção anterior 4.4. Na Tabela 8, os resultados deste modelo podem ser averiguados.

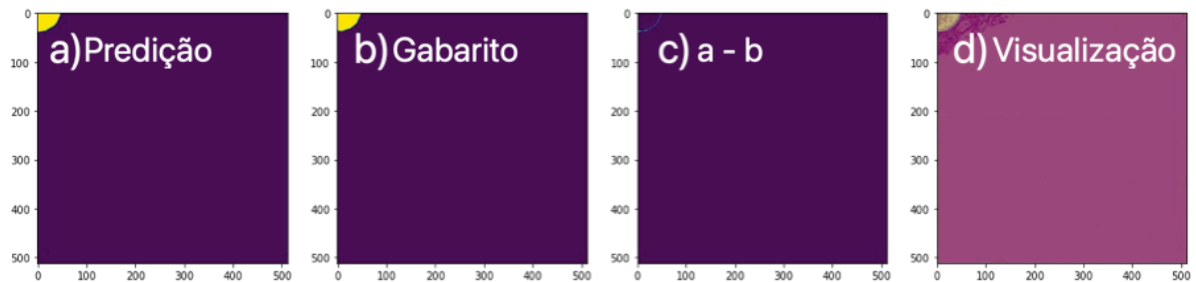
Tabela 8 - Resultados do modelo de atenção, *dataset* de validação

Arquitetura	DICE
Nenhuma	94,09%
SCSE	<b>94,22%</b>

Fonte: Autor

A partir destas análises, foi observado que a métrica de validação foi ligeiramente melhor que o experimento anterior. Veja-se na Figura 30.

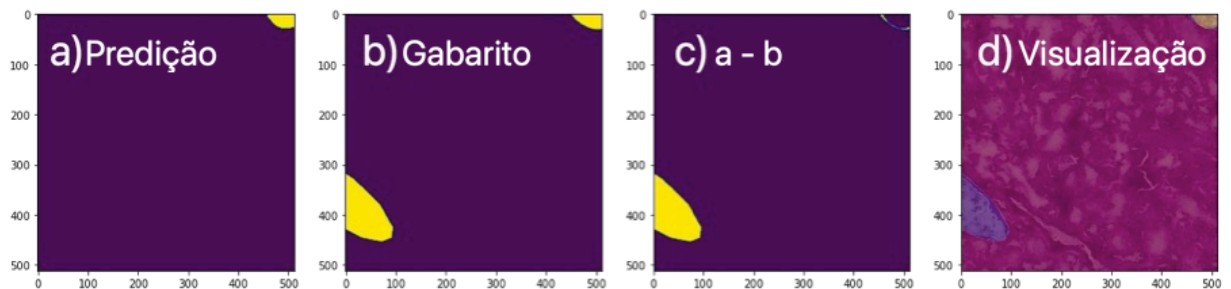
Figura 30 - Melhor caso do modelo de atenção, *dataset* de validação



Fonte: Autor

A Figura 30 apresenta o melhor caso produzido pelo modelo, com uma excelente segmentação, obtendo apenas alguns pixels de divergência em relação ao gabarito, como pode-se averiguar pela imagem (c), que representa a diferença entre o gabarito (b) e o *output* do modelo (a). Embora tal sucesso, a Figura 31 representa o pior caso.

Figura 31 - Pior caso do modelo de atenção, *dataset* de validação



Fonte: Autor

No pior caso, visualiza-se o mesmo exemplo do experimento anterior. A Figura 31 mostra que apesar da segmentação do glomérulo superior direito ter melhorado ligeiramente, o glomérulo posicionado no canto inferior esquerdo, segmentado em azul na imagem (d), foi ignorado totalmente pelo modelo. O ganho na *performance* DICE deste experimento foi de apenas 0,13%. Apesar deste modesto resultado, as camadas de atenção não apresentaram um aumento significativo no custo computacional do modelo, tornando este modelo seja o melhor encontrado por este trabalho.

#### 4.6 U-NET ++

O último experimento realizado, consistiu no treinamento de duas versões da U-Net++, sendo: com e sem a camada de atenção. Os resultados são apresentados na Tabela 9.

Tabela 9 - Resultados U-Net++, *dataset* de validação

Arquitetura	DICE
Nenhuma	93,94%
SCSE	<b>94,16%</b>

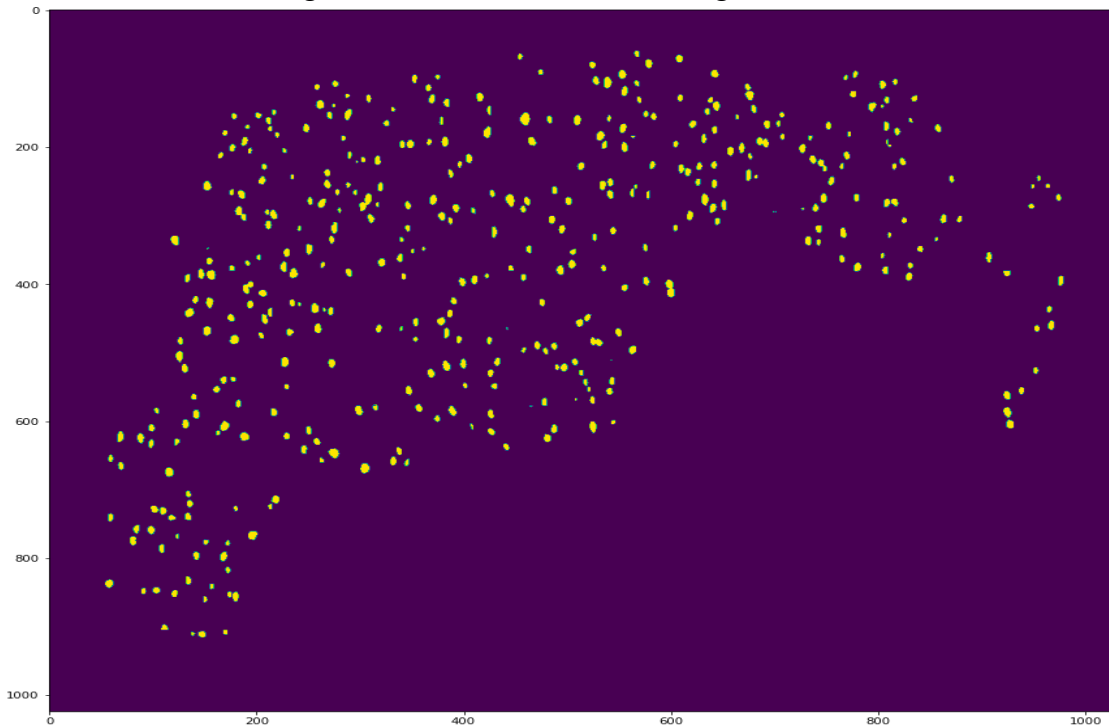
Fonte: Autor.

Observa-se na Tabela 9 que ambos os modelos, com e sem atenção, apresentam um resultado inferior ao apresentado no último experimento. Diante do exposto, manteve-se o modelo do experimento anterior.

#### 4.7 PERFORMANCE DO MODELO EM IMAGENS INTEIRAS

Até o momento esta pesquisa avaliou os modelos com base na *performance* DICE média obtida em recortes do *dataset* de validação. Outra maneira de avaliá-los é através da *performance* média obtida nas imagens de validação inteiras. O último método citado foi o utilizado pelos organizadores da competição para avaliar os modelos submetidos. Na Figura 32 é possível ver a máscara de segmentação gerada para uma imagem inteira do *dataset* de validação.

Figura 32 - Amostra totalmente segmentada



Fonte: Autor

Os resultados obtidos nas imagens inteiras pelo modelo que apresentou uma *performance* no dataset de validação de 94,22% na métrica DICE podem ser observados na Tabela 10.

Tabela 10 - Resultados nas imagens de validação

<b>Id da Imagem</b>	<b>DICE</b>
4ef6695ce	93,14%
b2dc8411c	94,98%
54f2eec69	92,28%
26dc41664	93,58%
0486052bb	92,50%

Fonte: Autor

A *performance* média do modelo no *dataset* de validação é, portanto, 93,29%, com um desvio padrão de 0,96%. A discrepância em relação a *performance* calculada a partir dos recortes (94,22%) é de 0,93%.

## 4.8 RESULTADOS DA COMPETIÇÃO HUBMAP - HACKING THE KIDNEY

A presente pesquisa foi submetida à competição *Hubmap - Hacking The Kidney*. por resultados, a pontuação obtida na competição foi de 92,33% na métrica DICE. Este resultado apresenta uma alta correlação entre o resultado obtido no conjunto de validação, apontando uma discrepância de apenas 0,96%, ou um desvio padrão. O *ranking* de pontuação dos participantes pode ser observado na Figura 33.

Figura 33 - *Leaderboard* publico da competição

**Leaderboard**

Public Private

This leaderboard is calculated with approximately 33% of the test data. The final results will be based on the other 67%, so the final standings may be different.

■ Prize Contenders

#	Team	Members	Score	Entries	Last	Code
1	MPWARE & GIBA [RAPIDS.AI]		0.9512	304	1y	
2	tom88jerry		0.9449	306	1y	
3	Chris Deotte [RAPIDS.AI]		0.9446	132	1y	
4 - 308		<a href="#">↕ See 304 More</a>				
309	<b>Felipe Sens Bonetto</b>		0.9233	20	1y	
310	Artyom Lyan		0.9232	10	1y	
311	Devesh Darshan		0.9231	14	1y	

Fonte: Adaptado de <https://www.kaggle.com/competitions/hubmap-kidney-segmentation/leaderboard?tab=public>

Diante do exposto, pode-se considerar que o resultado deste estudo é melhor do que o obtido por 74,6% dos outros 1216 participantes, ficando logo abaixo do primeiro quartil. Na Figura 33 verifica-se que o primeiro colocado obteve um score de 95,12%, apenas 2,79% maior que o resultado obtido por este trabalho.

## 5 CONCLUSÃO

Neste trabalho de conclusão de curso foi apresentado um modelo de inteligência artificial capaz de segmentar tecidos funcionais do rim, contribuindo com a base tecnológica necessária para auxiliar trabalhos futuros de análise em larga escala nas investigações sobre os glomérulos presentes no rim.

Para alcançar o modelo proposto, a etapa de treino foi efetuada por meio de longos procedimentos experimentais estruturados em busca da melhor *performance* de capacidade de segmentação utilizando redes neurais convolucionais. A relevância da pesquisa perfaz a necessidade na área médica na extração facilitada de informações clinicamente relevantes por meio de equipamentos e métodos eficientes, como os descritos neste estudo.

Atualmente, a identificação manual dos tecidos funcionais do rim é custosa e ineficiente exigindo tempo e esforço de patologistas treinados. Dessa forma, os modelos de inteligência artificial e *deep learning*, como os métodos das Redes Neurais Convolucionais utilizando a arquitetura U-Net e U-Net++ vem se mostrando um auxiliar promissor para os atuais métodos de identificação de glomérulos, permitindo um menor custo e maior precisão. Neste aspecto, o conhecimento da localização, tamanho e forma das diferentes estruturas anatômicas é um passo fundamental na compreensão e auxílio ao patologista para análise de imagens para exames e diagnóstico.

O modelo de segmentação semântica de imagens médicas, treinado neste trabalho de conclusão de curso foi capaz de atender aos requisitos para a submissão e análise de *performance* da competição *HuBMAP - Hacking the Kidney* ao qual foi submetido. As segmentações geradas apresentaram um resultado melhor do que 74% dos modelos, de outros participantes, submetidos à competição.

Tabela 11 - Comparação de resultados

<b>Métrica de Avaliação</b>	Ishrak; Khan; Imam (2021)	Govind et al. (2018)	<b>Autor</b>	<b>Ganhador da Competição HubMAP</b>
<b>DICE</b>	82,7%	83%	92,33%	<b>95,12%</b>

Fonte: Autor

Na Tabela 11 observa-se que os resultados, produzidos por essa pesquisa e pelos ganhadores da competição, superam os resultados previamente atingidos pelo estado da arte.

Isso corrobora com a pesquisa conduzida por Brown et al. (2020), que aponta que além da quantidade de dados a qualidade dos dados de treino é extremamente importante. Neste ponto, este estudo conclui que a qualidade e quantidade de dados superior do *dataset* disponibilizado por Allen J et al. (2021), em relação aos *datasets* utilizados em trabalhos anteriores, foi um fator decisivo para o bom resultado produzido por este trabalho.

Através de análises quantitativas expostas no capítulo 4 é possível verificar que a máscara obtida a partir da inferência captura a estrutura anatômica geral do glomérulo, apesar de falsos negativos estarem presentes. Os passos realizados durante o treinamento apresentaram uma melhora contínua e gradual do modelo, indicando que o processo de otimização foi bem-sucedido. O *score* promissor obtido na métrica DICE, com uma alta correlação entre o *dataset* de validação e a pontuação obtida na competição, assim como uma *performance* poucos pontos percentuais abaixo do primeiro colocado e os resultados qualitativos apresentados neste trabalho, sugerem que o objetivo de criar um modelo capaz de segmentar tecidos funcionais do rim é tecnologicamente viável, apresentando potencial tecnológico para desenvolvimento de uma tecnologia que auxilie o patologista em seu trabalho.

No entanto, em relação aos limitadores da pesquisa, bem como as sugestões para futuros pesquisadores, enfatiza que, é necessário avaliar a robustez do modelo em amostras provenientes de diversas etnias, idades, patologias e tamanhos antes de que se possa afirmar que o problema foi completamente solucionado. Além disso, o custo computacional elevado e o uso de aceleradores de *hardware* dedicados, são uma barreira para a implementação em larga escala desta tecnologia.



## REFERÊNCIAS

AHMED, Mohamed. **Medical Image Segmentation using Attention-Based Deep Neural Networks**. 2020. 50 f. TCC (Graduação) - Curso de Medical Engineering, Kth Royal Institute Of Technology School Of Engineering Sciences In Chemistry, Biotechnology And Health, Stockholm, Sweden, 2020. Disponível em: <https://www.diva-portal.org/smash/get/diva2:1477227/FULLTEXT01.pdf>. Acesso em: 30 jun. 2022.

ALAKE, Richmond. **What AlexNet Brought To The World Of Deep Learning**. 2020. Disponível em: <https://towardsdatascience.com/what-alexnet-brought-to-the-world-of-deep-learning-46c7974b46fc>. Acesso em: 25 jun. 2022

ALLEN J; ANDERSON D; BREWER M; CAPRIOLI R; DECAESTECKER M; FOGO A; GUTIERREZ D; HARRIS R; HARVEY J; MIGAS L; NEUMANN E; PATTERSON H; SHARMAN K; SPRAGGINS J; TIDEMAN L; VAN DE PLAS R; YANG H; SCHERSCHEL L; QUARDOKUS E; ... JU Y. **HuBMAP “Hacking the Kidney” 2021 Kaggle Competition Dataset - Glomerulus Segmentation on Periodic acid-Schiff Whole Slide Images**. 2021. DOI 10.35079/hbm925.sgx1.596. Disponível em: <https://portal.hubmapconsortium.org/browse/collection/4964d24bbc6668a72c4cbb5e0393a6bc>. Acesso em: 25 Jun. 2022.

APP · STREAMLIT. [s. d.]. Disponível em: <https://alumentations-demo.herokuapp.com/>. Acesso em: 25 Jun. 2022.

BAJAJ, Aayush. **Performance Metrics in Machine Learning**. 18 Mar. 2022. Disponível em: <https://neptune.ai/blog/performance-metrics-in-machine-learning-complete-guide>. Acesso em: 25 Jun. 2022.

BELL, Daniel; MOORE, Candace. **Dice similarity coefficient**, 15 Mar. 2020. DOI 10.53347/RID-75056. Disponível em: <http://radiopaedia.org/articles/75056>. Acesso em: 30 Jun. 2022.

BOHRA, Yash. **Vanishing and Exploding Gradients in Deep Neural Networks**. 2021. Disponível em: <https://www.analyticsvidhya.com/blog/2021/06/the-challenge-of-vanishing-exploding-gradients-in-deep-neural-networks/>. Acesso em: 26 Jun. 2022.

BROWN, Tom B.; MANN, Benjamin; RYDER, Nick; SUBBIAH, Melanie; KAPLAN, Jared; DHARIWAL, Prafulla; NEELAKANTAN, Arvind; SHYAM, Pranav; SASTRY, Girish; ASKELL, Amanda; AGARWAL, Sandhini; HERBERT-VOSS, Ariel; KRUEGER, Gretchen; HENIGHAN, Tom; CHILD, Rewon; RAMESH, Aditya; ZIEGLER, Daniel M.; WU, Jeffrey; WINTER, Clemens; ... AMODEI, Dario. **Language Models are Few-Shot Learners**. In: **ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS**, 33., 2020, Vancouver. **Language Models are Few-Shot Learners**. Vancouver: Nips, 2020. p. 1-25. Disponível em: <https://arxiv.org/abs/2005.14165>. Acesso em: 25 jun. 2022.

BUSLAEV, Alexander; IGLOVIKOV, Vladimir I.; KHVEDCHENYA, Eugene; PARINOV, Alex; DRUZHININ, Mikhail; KALININ, Alexandr A.. **Alumentations: fast and flexible image augmentations**. **Information**, [S.L.], v. 11, n. 2, p. 125, 24 fev. 2020. MDPI AG. <http://dx.doi.org/10.3390/info11020125>.

CÂMARA, Brunno. **Entendendo o que é filtração, reabsorção, secreção e excreção renal.** 23 Dec. 2013. Disponível em: <https://www.biomedicinapadrao.com.br/2013/12/entendendo-o-que-e-filtracao-reabsorcao.html>. Acesso em: 26 Jun. 2022.

CHEN, Jiasi; RAN, Xukan. Deep Learning With Edge Computing: a review. **Proceedings Of The Ieee**, [S.L.], v. 107, n. 8, p. 1655-1674, ago. 2019. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/jproc.2019.2921977>.

CUNHA, Leonardo Cardoso da. **REDES NEURAIIS CONVOLUCIONAIS E SEGMENTAÇÃO DE IMAGENS-UMA REVISÃO BIBLIOGRÁFICA.** 2020. 55 f. TCC (Graduação) - Curso de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2020. Disponível em: [https://www.monografias.ufop.br/bitstream/35400000/2872/6/MONOGRAFIA\\_RedNeuraisConvolutionais.pdf](https://www.monografias.ufop.br/bitstream/35400000/2872/6/MONOGRAFIA_RedNeuraisConvolutionais.pdf). Acesso em: 25 jun. 2022.

BONO, Bernard de; GRENON, Pierre; BALDOCK, Richard; HUNTER, Peter. Functional tissue units and their primary tissue motifs in multi-scale physiology. **Journal Of Biomedical Semantics**, [S.L.], v. 4, n. 1, p. 22, 2013. Springer Science and Business Media LLC. <http://dx.doi.org/10.1186/2041-1480-4-22>

GINLEY, Brandon; TOMASZEWSKI, John E.; YACOUB, Rabi; CHEN, Feng; SARDER, Pinaki. Unsupervised labeling of glomerular boundaries using Gabor filters and statistical testing in renal histology. **Journal Of Medical Imaging**, [S.L.], v. 4, n. 2, p. 021102, 28 fev. 2017. SPIE-Intl Soc Optical Eng. <http://dx.doi.org/10.1117/1.jmi.4.2.021102>.

GIRMA, Abenezer. **Convolutional Neural Network in a Nutshell.** 2019. Disponível em: <https://abenezer-g.medium.com/part-1-convolutional-neural-network-in-a-nutshell-89f81a329ec3>. Acesso em: 26 Jun. 2022.

GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. **Deep Learning.** Massachusetts: Mit Press, 2016.

GINLEY, Brandon; LUTNICK, Brendon; TOMASZEWSKI, John E.; SARDER, Pinaki; GOVIND, Darshana. Glomerular detection and segmentation from multimodal microscopy images using a Butterworth band-pass filter. **Medical Imaging 2018: Digital Pathology**, [S.L.], v. 10581, n. 1, p. 297-303, 6 mar. 2018. SPIE. <http://dx.doi.org/10.1117/12.2295446>

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep Residual Learning for Image Recognition. **Proceedings Of The Ieee Computer Society Conference On Computer Vision And Pattern Recognition**, [S.L.], v. 2016, n. 1, p. 770-778, 2015. ArXiv. <http://dx.doi.org/10.48550/ARXIV.1512.03385>.

ISHRAK, Sheikh Sakib; KHAN, Sakif; IMAM, Ahmad. **Semantic Segmentation of Glomeruli in Human Kidney Tissue Images.** 2020. 56 f. TCC (Graduação) - Curso de Computer Science And Engineering, Islamic University Of Technology, Dhaka, 2020. Disponível em: [http://103.82.172.44:8080/xmlui/bitstream/handle/123456789/1310/Thesis2020\\_CSE\\_160041054\\_160041038\\_160041039.PDF%20-](http://103.82.172.44:8080/xmlui/bitstream/handle/123456789/1310/Thesis2020_CSE_160041054_160041038_160041039.PDF%20-)

%20Ahmad%20Imam%20%2c160041054.pdf?sequence=1&isAllowed=y. Acesso em: 25 jun. 2022.

JHA, Debesh; RIEGLER, Michael A.; JOHANSEN, Dag; HALVORSEN, Pål; JOHANSEN, Håvard D.. DoubleU-Net: a deep convolutional neural network for medical image segmentation. **Proceedings - Ieee Symposium On Computer-Based Medical Systems**, [S.L.], v. 1, n. 1, p. 558-564, 2020. ArXiv. <http://dx.doi.org/10.48550/ARXIV.2006.04868>.

**KAGGLE: YOUR HOME FOR DATA SCIENCE**. 2022. Disponível em: <https://www.kaggle.com/>. Acesso em: 26 Jun. 2022.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P.. Gradient-based learning applied to document recognition. **Proceedings Of The Ieee**, [S.L.], v. 86, n. 11, p. 2278-2324, 1998. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/5.726791>

MALACH, Eran; SHALEV-SHWARTZ, Shai. Computational Separation Between Convolutional and Fully-Connected Networks. **Arxiv**, [S.L.], p. 1-13, 2020. ArXiv. <http://dx.doi.org/10.48550/ARXIV.2010.01369>.

MCCULLOCH, Warren; PITTS, Walter. A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITIES. **Bulletin Of Mathematical Biophysics**, Londres, v. 5, n. 1, p. 115-133, jun. 1943. Disponível em: <https://www.cse.chalmers.se/~coquand/AUTOMATA/mcp.pdf>. Acesso em: 25 jun. 2022.

NUNES, Ivan; DANILO, Silva ; SPATTI, Hernane; ANDRADE, Rogerio; LUISA, Flauzino; LIBONI, Helena Bartocci; FRANCO, Silas; ALVES, Reis. **Artificial Neural Networks A Practical Course**. [S. l.: s. n.], 2017.

PASZKE, Adam; GROSS, Sam; MASSA, Francisco; LERER, Adam; BRADBURY GOOGLE, James; CHANAN, Gregory; KILLEEN, Trevor; LIN, Zeming; GIMELSHEIN, Natalia; ANTIGA, Luca; DESMAISON, Alban; XAMLA, Andreas Köpf; YANG, Edward; DEVITO, Zach; RAISON NABLA, Martin; TEJANI, Alykhan; CHILAMKURTHY, Sasank; AI, Qure; STEINER, Benoît; ... CHINTALA, Soumith. **PyTorch: An Imperative Style, High-Performance Deep Learning Library**. 2019. . Acesso em: 26 Jun. 2022.

POUDEL, Rudra P.K.; LIWICKI, Stephan; CIPOLLA, Roberto. Fast-SCNN: Fast Semantic Segmentation Network. **30th British Machine Vision Conference 2019, BMVC 2019**, 12 Feb. 2019. DOI 10.48550/arxiv.1902.04502. Disponível em: <https://arxiv.org/abs/1902.04502v1>. Acesso em: 30 Jun. 2022.

RAMSUNDAR, Bharath; ZADEH, Reza Bosagh. 4. **Fully Connected Deep Networks - TensorFlow for Deep Learning**. 2018. Disponível em: <https://www.oreilly.com/library/view/tensorflow-for-deep/9781491980446/ch04.html>. Acesso em: 26 Jun. 2022.

RODRIGUES, Diego Alves. **DEEP LEARNING E REDES NEURAIAS CONVOLUCIONAIS: RECONHECIMENTO AUTOMÁTICO DE CARACTERES EM PLACAS DE LICENCIAMENTO AUTOMOTIVO**. 2018. 37 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Informática Universidade Federal da Paraíba,

João Pessoa, 2018. Disponível em:

<https://repositorio.ufpb.br/jspui/bitstream/123456789/15606/1/DAR20052019.pdf>. Acesso em: 25 jun. 2022.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. 2015-U-Net. **arXiv**, , p. 1–8, 2015. Disponível em: <http://lmb.informatik.uni-freiburg.de/%0Aarxiv:1505.04597v1>. Acesso em: 30 Jun. 2022.

ROY, Abhijit Guha; NAVAB, Nassir; WACHINGER, Christian. Concurrent Spatial and Channel Squeeze & Excitation in Fully Convolutional Networks. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, vol. 11070 LNCS, p. 421–429, 7 Mar. 2018. DOI 10.48550/arxiv.1803.02579. Disponível em: <https://arxiv.org/abs/1803.02579v2>. Acesso em: 30 Jun. 2022.

SAMUEL, Terry; HOY, Wendy E.; DOUGLAS-DENTON, Rebecca; HUGHSON, Michael D.; BERTRAM, John F. Applicability of the glomerular size distribution coefficient in assessing human glomerular volume: the Weibel and Gomez method revisited. **Journal of Anatomy**, vol. 210, no. 5, p. 578, May 2007. DOI 10.1111/J.1469-7580.2007.00715.X. Disponível em: </pmc/articles/PMC2375741/>. Acesso em: 30 Jun. 2022.

SARDER, Pinaki; GINLEY, Brandon; TOMASZEWSKI, John E. **Automated renal histopathology: Digital extraction and quantification of renal pathology**. 2016. DOI 10.1117/12.2217329. Disponível em: <http://proceedings.spiedigitallibrary.org/>. Acesso em: 26 Jun. 2022.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. **3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings**, 4 Sep. 2014. DOI 10.48550/arxiv.1409.1556. Disponível em: <https://arxiv.org/abs/1409.1556v6>. Acesso em: 26 Jun. 2022.

TAN, Mingxing; LE, Quoc v. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. **36th International Conference on Machine Learning, ICML 2019**, vol. 2019-June, p. 10691–10700, 28 May 2019. DOI 10.48550/arxiv.1905.11946. Disponível em: <https://arxiv.org/abs/1905.11946v5>. Acesso em: 26 Jun. 2022.

VIEIRA, Rafael. Histologia do rim - **Néfron, glomérulo, córtex e medula**. 6 Jan. 2022. Disponível em: <https://www.kenhub.com/pt/library/anatomia/histologia-do-rim>. Acesso em: 25 Jun. 2022.

WALCZAK, Steven. **Artificial Neural Networks**. , p. 120–131, 2019. DOI 10.4018/978-1-5225-7368-5.CH004: Disponível em: <http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-5225-7368-5.ch004>. Acesso em: 26 Jun. 2022.

WANG, Pei; MONETT, Dagmar; LEWIS, Colin W P; THÓRISSON, Kristinn R. On Defining Artificial Intelligence. **Journal of Artificial General Intelligence**, vol. 10, no. 2, p. 2019–2022, 2019. <https://doi.org/10.2478/jagi-2019-0002>. Acesso em: 26 Jun. 2022.

WIDROW BERNARD; HOFF MARCIAN E. **Associative Storage and Retrieval of Digital Information in Networks of Adaptive Neurons**. 1962. Disponível em: <https://isl.stanford.edu/~widrow/papers/c1962associativestorage.pdf>. Acesso em: 26 Jun. 2022.

YAKUBOVSKIY, Pavel. **Segmentation Models Pytorch**. 2020. Disponível em: [https://github.com/qubvel/segmentation\\_models.pytorch](https://github.com/qubvel/segmentation_models.pytorch). Acesso em: 26 Jun. 2022.

YERRAM, Varun. **Understanding DICE COEFFICIENT**. 26 Dec. 2020. Disponível em: <https://www.kaggle.com/code/yerramvarun/understanding-dice-coefficient/notebook>. Acesso em: 25 Jun. 2022.

ZAMAN KHAN, Rafiqul; ALLAMY, haider; HONG, Wei; KHALAF JABBAR RAFIQU L ZAMAN KHAN, Haider DR. METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY) Related papers Neural Net works for Classification: **A Survey METHODS TO AVOID OVER-FITTING AND UNDER-FITTING IN SUPERVISED MACHINE LEARNING (COMPARATIVE STUDY)**. 2015. . Acesso em: 26 Jun. 2022.

ZEDALIS, Julianne; EGGBRECHT, John; OPENSTAX COLLEGE; OPEN TEXTBOOK LIBRARY. **Biology for AP® Courses**. , p. 1393–1437, 2018. . Acesso em: 30 Jun. 2022.

ZHANG, Jeremy. **UNet — Line by Line Explanation**. 2019. Disponível em: <https://towardsdatascience.com/unet-line-by-line-explanation-9b191c76baf5>. Acesso em: 26 Jun. 2022.

ZHOU, Zongwei; RAHMAN SIDDIQUEE, Md Mahfuzur; TAJBAKSH, Nima; LIANG, Jianming. UNet++: A Nested U-Net Architecture for Medical Image Segmentation. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, vol. 11045 LNCS, p. 3–11, 18 Jul. 2018. DOI 10.48550/arxiv.1807.10165. Disponível em: <https://arxiv.org/abs/1807.10165v1>. Acesso em: 30 Jun. 2022.