UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Roberto Augusto Philippi Martins

Análise do método de aprendizado semi-supervisionado Teacher-Student para segmentação de imagens médicas usando redes neurais convolucionais

Florianópolis
2022

Roberto Augusto Philippi Martins

Análise do método de aprendizado semi-supervisionado Teacher-Student para segmentação de imagens médicas usando redes neurais convolucionais

Dissertação submetida ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina para a obtenção do título de mestre em Engenharia Elétrica.
Orientador: Prof. Danilo Silva, Ph.D.

Florianópolis
2022

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Martins, Roberto Augusto Philippi
Análise do método de aprendizado semi-supervisionado Teacher-Student para segmentação de imagens médicas usando redes neurais convolucionais / Roberto Augusto Philippi Martins ; orientador, Danilo Silva, 2022.
64 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico, Programa de Pós-Graduação em Engenharia Elétrica, Florianópolis, 2022.

Inclui referências.

1. Engenharia Elétrica. 2. Aprendizado semi supervisionado. 3. Segmentação semântica. 4. Teacher Student. 5. Diagnóstico assistido por computador. I. Silva, Danilo . II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Roberto Augusto Philippi Martins

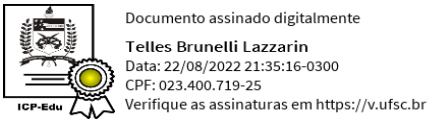
Análise do método de aprendizado semi-supervisionado Teacher-Student para segmentação de imagens médicas usando redes neurais convolucionais

O presente trabalho em nível de mestrado foi avaliado e aprovado por banca examinadora composta pelos seguintes membros:

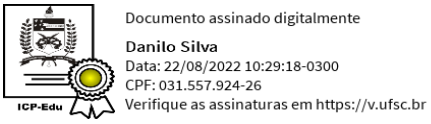
Prof. Jefferson Luiz Brum Marques, Ph.D.
Universidade Federal de Santa Catarina

Prof. Marcelo Ricardo Stemmer, Dr.
Universidade Federal de Santa Catarina

Certificamos que esta é a **versão original e final** do trabalho de conclusão que foi julgado adequado para obtenção do título de mestre.



Prof. Telles Brunelli Lazzarin, Dr.
Coordenador do Programa de
Pós-Graduação em Engenharia Elétrica



Prof. Danilo Silva, Ph.D.
Orientador

Florianópolis, 2022.

Dedico este trabalho à Kassia.

AGRADECIMENTOS

Agradeço à minha família por todo o apoio que recebi durante essa jornada. Em especial, agradeço minha mãe Renita, e minhas irmãs Barbara e Elisabeth, por sempre acreditarem em mim e me apoiarem de diversas formas.

Um agradecimento especial à Kassia, por estar ao meu lado durante todo esse tempo e me incentivar a concluir este trabalho. Agradeço a todo o carinho e atenção, e pelas inúmeras correções feitas neste texto.

Aos amigos da graduação e aos colegas de casa pela companhia e parceria durante os últimos anos.

Agradeço ao professor Danilo Silva por me acompanhar desde a graduação, pelas ideias e correções durante o desenvolvimento desse trabalho.

Aos professores Jefferson Luiz Brum Marques e Marcelo Ricardo Stemmer por aceitarem participar da banca avaliadora desta dissertação.

À toda universidade, professores, pesquisadores e colegas, por todas as contribuições que permitiram a execução desta pesquisa.

RESUMO

Um dos maiores desafios encontrados na área de aprendizado de máquina é a obtenção de conjuntos de dados suficientemente grandes para realização do treinamento de redes neurais. A obtenção de amostras anotadas é especialmente difícil dentro da área de biomédica, dado que as imagens de interesse tem o acesso limitado devido a questões técnicas e legais, e visto que a anotação correta das amostras depende do trabalho de profissionais qualificados. Aprendizado Semi-Supervisionado (SSL) é uma técnica que permite reduzir a dependência em amostras anotadas, uma vez que ela consegue fazer uso de amostras não anotadas junto ao treinamento de redes neurais. Neste trabalho é avaliada a utilização do método de aprendizado semi-supervisionado Teacher-Student no treinamento de um modelo de segmentação semântica, voltado para a avaliação de lesões em imagens de exames de fundo de olho para diagnóstico de retinopatia diabética. É observado que a utilização desse método permite o treinamento bem sucedido de um modelo, utilizando um conjunto de dados extremamente reduzido, com um ganho de performance significativo quando comparado ao modelo treinado de maneira supervisionada. Obtemos *AUPR* = 0,587 com o uso de SSL quando utilizando apenas 16 imagens do conjunto de dados FGADR, comparado com *AUPR* = 0,513 sem a utilização de SSL. Adicionalmente, quando utilizando adaptação de domínio com o conjunto de dados IDRiD, observamos um aumento de *AUPR* = 0,542 para *AUPR* = 0,594 com o uso de SSL.

Palavras-chave: Aprendizado semi-supervisionado. Segmentação semântica. Teacher-Student. Diagnóstico Assistido por Computador.

ABSTRACT

One of the biggest challenges found in the field of machine learning is to obtain large enough datasets in order to train neural network models. Obtaining annotated samples is especially difficult within the biomedical field, since access to medical data is often hindered by technical and legal barriers, and since the correct annotation of samples requires the work of qualified professionals. Semi-Supervised Learning (SSL) is a technique that reduces the dependence on annotated samples, since it manages to incorporate unannotated samples in the supervised training of neural networks. This work evaluates the use of the Teacher-Student semi-supervised learning method in the training of a semantic segmentation model, aimed at the evaluation of lesions in images of eye fundus exams for the diagnosis of diabetic retinopathy. It is observed that the use of this method allows the successful training of a model using an extremely small dataset, with a significant performance gain when compared to the purely supervised counterpart. We show an improvement on the performance from *AUPR* = 0.513 to *AUPR* = 0.587 when using SSL and training with only 16 images from the FGADR dataset. Additionally, when using domain adaptation with the IDRiD dataset, we show an increase from *AUPR* = 0,542 to *AUPR* = 0,594 with the use of SSL.

Keywords: Semi-supervised Learning. Semantic segmentation. Teacher-Student. Computer Aided Diagnosis.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

SSL	Semi-Supervised Learning
FGADR	Fine-Grained Annotated Diabetic Retinopathy
IDRID	Indian Diabetic Retinopathy Image Dataset
CAD	Computer Aided Diagnosis
RD	Retinopatia Diabética
OCR	Optical Character Recognition
GAN	Generative Adversarial Network
KL – Divergence	Kullback–Leibler Divergence
ReLU	Rectified Linear Unit
AUPR	Area Under Precision Recall curve
AUROC	Area Under Receiver Operating Characteristic
SDI	Sørensen–Dice Index
CE	Cross-Entropy
WBCE	Weighted Binary Cross-Entropy
KD	Knowledge Distillation
EMA	Exponential Moving Average
MSE	Mean Square Error
LR	Learning Rate

LISTA DE SÍMBOLOS

$\mathcal{D}$	Conjunto de dados anotado
$\mathcal{U}$	Conjunto de dados não anotado
$\mathcal{L}_{\mathcal{D}}$	Conjunto de anotações referentes à $\mathcal{D}$
$\mathcal{P}_{\mathcal{D}}$	Conjunto de predições sobre $\mathcal{D}$
$\mathcal{P}_{\mathcal{U}}$	Conjunto de predições sobre $\mathcal{U}$
K	Parâmetro que define o nível de rigidez na definição do intervalo de confiança na geração de pseudo labels
P	Parâmetro que define a combinação de imagens anotadas e pseudo labels no treinamento do modelo Student
L_{CE}	Perda Entropia Cruzada
L_{Dice}	Perda Dice

SUMÁRIO

1 INTRODUÇÃO

Nos últimos anos o desenvolvimento de ferramentas de diagnóstico auxiliado por computador (computer-aided diagnosis — CAD) tem sido acelerado pela inclusão de técnicas de aprendizado de máquina (ZHAO, D. *et al.*, 2020). O uso de redes convolucionais tem se mostrado o estado da arte em diversas funções de visão computacional, inclusive no auxílio ao diagnóstico de imagens médicas.

Atualmente, o principal desafio encontrado no desenvolvimento destes algoritmos está na obtenção de imagens médicas com laudos de alta qualidade para o treinamento dos modelos. O treinamento adequado e performance obtida com uma rede convolucional são diretamente ligados ao número de imagens disponíveis para esse processo de otimização (OQUAB *et al.*, 2014).

Imagens médicas contê uma grande diversidade intra-domínio, ou seja, é comum observar variações estatísticas significativas mesmo sobre imagens relacionadas (GUAN; LIU, 2022). Para gerar um modelo robusto capaz de trabalhar com essas variações, é necessário ter-se um número adequado de exemplos de treinamento, a fim de varrer uma área suficientemente grande do espaço de observação.

A obtenção de imagens e laudos médicos requer o trabalho de profissionais especializados na área para geração de amostras confiáveis e de qualidade, um processo demorado que consiste na anotação manual e detalhada de exames médicos, sendo que até mesmo profissionais qualificados podem discordar na geração de anotações (ALGAN *et al.*, 2020; KALPATHY-CRAMER *et al.*, 2016). A obtenção de conjuntos de dados se torna ainda mais difícil devido a barreiras técnicas e legais, como o acesso à bancos de dados de instituições médicas e uso de dados pessoais.

Por outro lado, o acesso a conjuntos de dados não anotados se mostra mais simples, devido a não requerer o diagnóstico médico feito por profissionais especializados para geração de laudos. Esse conjunto não pode ser utilizado para o treinamento supervisionado de uma rede neural, devido a falta de anotações, mas a informação contida nas próprias imagens pode ser utilizada de forma auxiliar para o treinamento de modelos mais robustos.

Aprendizado Semi-Supervisionado (SSL) é uma abordagem de aprendizado de máquina que permite incorporar um conjunto não anotado dentro do treinamento supervisionado de uma rede neural. Essa técnica tem como objetivo produzir um modelo com melhor performance quando comparado com a contrapartida puramente supervisionada, e é utilizada principalmente em problemas onde a obtenção de amostras é fácil quando comparado à obtenção de anotações (CHAPELLE; SCHÖLKOPF; ZIEN, 2006).

Neste trabalho analisa-se a aplicação de aprendizado semi-supervisionado baseado numa arquitetura Teacher-Student (YALNIZ *et al.*, 2019; LEE, 2013; XIE *et al.*,

2020) para o treinamento de redes convolucionais profundas. Esse método realiza o treinamento do modelo em um processo de múltiplas etapas, fazendo uso de um par de modelos (chamados de Teacher e Student) e de pseudo labels para conseguir incorporar amostras não anotadas no treinamento da rede.

Para a avaliação do algoritmo, buscou-se um domínio com imagens médicas onde a segmentação de lesões faz parte do processo de diagnóstico. Escolheu-se aplicar o método para a segmentação de lesões em exames de fundo de olho utilizados para diagnóstico de retinopatia diabética, utilizando o conjunto de dados *Fine-Grained Annotated Diabetic Retinopathy* (FGADR) (ZHOU *et al.*, 2020) para treinamento e avaliação do modelo.

Esse domínio e conjunto de dados foram escolhidos devido as suas características que permitem a avaliação desse método. Trata-se de um conjunto de imagens que contém anotações na forma de máscaras binárias para diferentes lesões nos exames e um número adequado de amostras para realização dos experimentos.

1.1 OBJETIVOS

1.1.1 Objetivos Gerais

Analisar o método de aprendizado semi-supervisionado Teacher-Student no treinamento de modelos convolucionais profundos para a segmentação de lesões em imagens médicas.

1.1.2 Objetivos Específicos

- Analisar o método de aprendizado semi-supervisionado Teacher-Student proposto por Yalniz et al. (YALNIZ *et al.*, 2019).
- Adaptar o método Teacher-Student para aplicações de segmentação semântica, permitindo o treinamento dos modelos e geração de pseudo labels para esse tipo de tarefa.
- Avaliar o desempenho do algoritmo aplicado à segmentação de imagens médicas, em específico no conjunto de dados FGADR (*Fine-Grained Annotated Diabetic Retinopathy*).

1.2 ORGANIZAÇÃO DO TEXTO

Capítulo 2 contém a fundamentação teórica necessária para leitura deste trabalho, onde é feita uma breve introdução à conceitos de aprendizado de máquina, visão computacional, segmentação semântica e redes residuais, bem como uma descrição do domínio utilizado neste trabalho.

No Capitulo 3 descrevemos em mais detalhes a área de aprendizado semi-supervisionado, comparando os principais métodos encontrados na literatura.

No Capitulo 4 é descrita a metodologia utilizada para a realização dos experimentos. Além do prório conjunto de dados, são detalhados o algoritmo Teacher-Student, a geração e processamento de pseudo labels, e os métodos utilizados para o treinamento dos modelos.

O Capítulo 5 detalha os experimentos e resultados obtidos. É feita uma descrição dos detalhes de treinamento, com os parâmetros utilizados para a otimização dos modelos. É demonstrada também a performance obtida para os diferentes experimentos realizados.

No Capítulo 6 faremos uma análise das contribuições e resultados encontrados nesse trabalho, bem como perspectivas futuras para pesquisa nessa área.

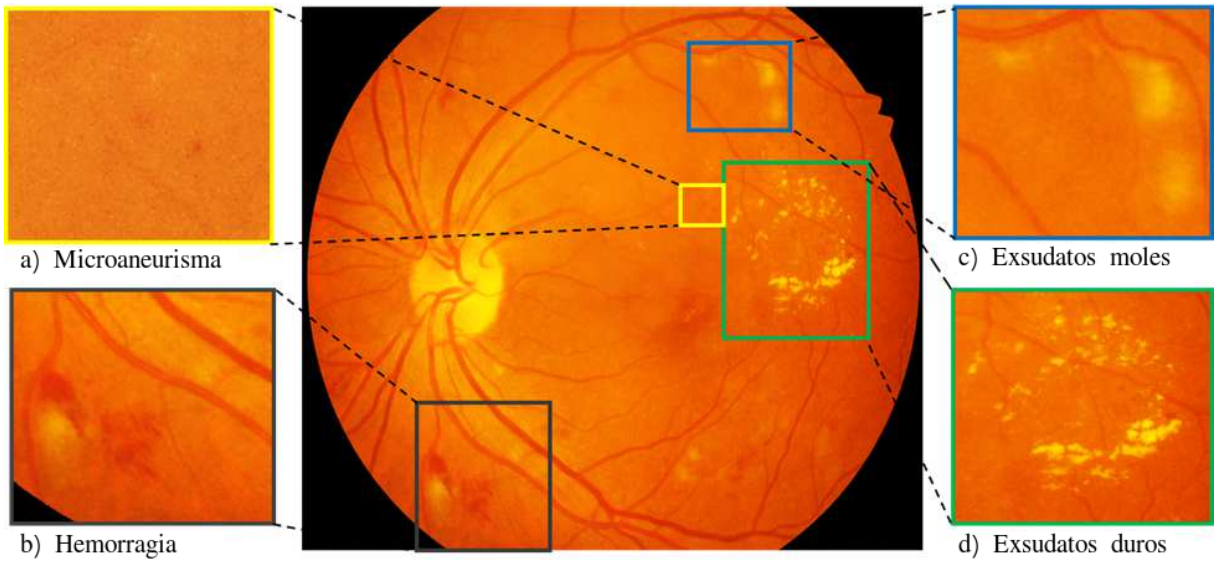
2 FUNDAMENTAÇÃO TEÓRICA

2.1 RETINOPATIA DIABÉTICA

O diabetes mellitus está relacionado com vários problemas de saúde, entre eles falha renal, doenças cardiovasculares e neuropatia. Nos olhos, o descontrole da glicemia pode causar danos aos vasos sanguíneos finos da retina e a proliferação de vasos anômalos, resultando na complicação conhecida como retinopatia diabética (MONTEIRO, 2020).

A retinopatia diabética (RD) é uma das principais causas de cegueira em adultos, presente em 22% dos indivíduos com diabetes, totalizando mais de 100 milhões de casos no mundo (TEO *et al.*, 2021). Os principais sintomas encontrados são microaneurismas, hemorragia intrarretiniana, exsudatos, edema e descolamento da retina, causando diferentes níveis de perda de visão dependendo do estágio da doença.

Figura 1 – Exemplo de um exame de fundo de olho com diferentes sintomas de retinopatia diabética marcados na imagem (PORWAL *et al.*, 2018). As áreas realçadas da imagem indicam a presença de microaneurisma, hemorragia, exsudatos moles e exsudatos duros.



O estágio inicial da doença é conhecido como retinopatia diabética não proliferativa, onde o paciente sofre pouca ou nenhuma perda de visão e observamos complicações mais leves da doença (MEHTA, 2020). Os principais sintomas neste estágio da doença são os microaneurismas, hemorragias e exsudatos moles e duros na retina posterior.

Em um estágio mais avançado da doença, conhecido como retinopatia diabética proliferativa, o paciente sofre com a complicação dos sintomas, aparecimento de novos vasos sanguíneos suscetíveis a rompimento na superfície da retina e ocorrência de

edema macular (MEHTA, 2020). Esse estágio da doença acompanha maior perda de visão e até mesmo cegueira, com cerca de metade dos pessoas nesse estágio da doença sofrendo com edema macular diabético.

A detecção da retinopatia diabética é feita primariamente com o uso de exames de fundo de olho, como mostra a Figura 1, também conhecido como fundoscopia ou oftalmoscopia. Esse procedimento é feito com a utilização de um oftalmoscópio e dilatação da pupila, e consiste no mapeamento da retina e outras áreas do seguimento posterior do olho, servindo para analisar as condições das artérias, veias e nervos da retina.

A detecção e o tratamento precoce da retinopatia diabética é essencial para prevenir casos avançados da doença. Os sintomas da doença tendem a complicar com o tempo, especialmente em pacientes com diabetes não tratada, e apesar de não ter cura é possível evitar a progressão dos sintomas com tratamentos adequados, como fotocoagulação da retina e injeção de medicações antiangiogênicas e corticosteróides (MEHTA, 2020).

2.2 VISÃO COMPUTACIONAL

A área de visão computacional (*computer vision*) tem como objetivo gerar algoritmos capazes de processar, analisar e identificar imagens de maneira similar ao processo realizado pelo ser humano.

O trabalho de visão computacional vai além do processamento de imagens, visando não só modificar as características da imagem, mas entender o seu conteúdo. As tarefas realizadas envolvem a interpretação do sinal, adquirindo e processando informações relevantes, produzindo então uma caracterização de alto nível da imagem original.

As principais tarefas de interesse realizadas na área de visão computacional podem ser divididas em:

- Classificação: categorizar a imagem dentro de um conjunto de possíveis classes conhecidas.
- Detecção de objetos: encontrar instâncias de objetos em uma imagem. Além da classificação dos objetos, é necessário também localizar os objetos dentro da imagem.
- Segmentação semântica: classificar individualmente os pixels de uma imagem dentre o conjunto de possíveis classes conhecidas.
- Segmentação de instâncias: segmentação de imagens que separa cada ocorrência de objetos dentro de uma classe. Cada instância de uma classe é segmentada de maneira independente.

Essas técnicas podem ser aplicadas em diversas áreas, gerando um sistema capaz de interpretar o mundo real e que permite a interação entre computadores e o ambiente. Algumas das áreas de aplicação de visão computacional são:

- Automação (*machine vision*, controle industrial, inspeção de produtos e mercadorias).
- Navegação (carros autônomos, prevenção de acidentes).
- Identificação de pessoas (controle de tráfego, vigilância)
- Leitura automática de textos (OCR).
- Diagnóstico médico (detecção de anomalias, segmentação de órgãos, classificação de exames, triagem automatizada).

2.3 SEGMENTAÇÃO SEMÂNTICA

Segmentação semântica é uma das principais áreas do campo de Visão Computacional, com objetivo de classificar individualmente os pixels de uma imagem entre as classes conhecidas. Diferente de outras tarefas como classificação de imagens, a segmentação semântica retorna uma informação muito mais detalhada sobre a imagem no formato de uma máscara de segmentação. Esse formato de predição mais detalhado permite que algoritmos de segmentação sejam utilizados para solucionar problemas na área de visão computacional, onde é necessário a localização precisa de objetos dentro da imagem.

Atualmente os principais métodos de segmentação semântica são baseados em redes convolucionais profundas (SULTANA; SUFIAN; DUTTA, 2020), como FCN (LONG; SHELHAMER; DARRELL, 2014), Mask R-CNN (HE *et al.*, 2017) e arquiteturas Encoder-Decoder (Codificador-Decodificador) (RONNEBERGER; FISCHER; BROX, 2015; CHAURASIA; CULURCIELLO, 2017). Esses modelos são construídos de maneira puramente convolucional, e permitem um treinamento *end-to-end* utilizando o algoritmo de *backpropagation*.

O trabalho de Ronneberger et al. (RONNEBERGER; FISCHER; BROX, 2015) demonstra a utilização da rede U-Net, baseada na arquitetura Encoder-Decoder, aplicado à segmentação de imagens médicas, e é uma das principais referências no desenvolvimento de redes convolucionais para segmentação semântica.

Nesse tipo de configuração, o Encoder é a parte do modelo responsável por codificar o sinal de entrada para um espaço latente, e o Decoder é responsável por mapear essa representação em uma máscara com a mesma resolução que a imagem original, contendo a categorização pixel a pixel da amostra.

Outros trabalhos aprimoram a arquitetura U-Net a partir da utilização de encoders e decoders mais poderosos, bem como com o desenvolvimento de métodos para otimizar a relação entre informação semântica e espacial nas máscaras produzidas.

O trabalho de Jegou et al. (JÉGOU *et al.*, 2016) demonstra a utilização de DenseNets como base para o Encoder do modelo, e Drozdzal et al. (DROZDZAL *et al.*, 2016) avalia a utilização de ResNets. Ambos os trabalhos alcançaram resultados superiores à utilização de blocos convolucionais simples, mostrando a importância da utilização de arquiteturas convolucionais profundas e mais poderosas.

Um dos principais desafios encontrados em redes de segmentação do tipo Encoder-Decoder é a perda de informação espacial durante o processamento do Encoder (CHEN, L. *et al.*, 2017). A utilização de redes convolucionais tradicionais nesse componente resulta na perda de informação espacial, já que um dos objetivos desses modelos é reduzir a dimensionalidade do sinal para a dimensão do espaço latente.

Outros trabalhos utilizam arquiteturas específicas no modelo do Encoder com fim a amenizar a perda de informação espacial, como Redes Neurais Recorrentes (SALVADOR *et al.*, 2017) e Conditional Random Fields (KRÄHENBÜHL; KOLTUN, 2012). Arquiteturas Encoder-Decoder minimizam este problema a partir da utilização de Conexões de Atalho (Skip Connections) (DROZDZAL *et al.*, 2016).

2.3.1 Skip connections

Ao saber que a recuperação da informação espacial a partir da representação obtida pelo Encoder é um processo difícil, devido à redução de dimensionalidade do sinal, busca-se simplificar o processamento do Decoder a partir da inclusão de informação proveniente diretamente do Encoder.

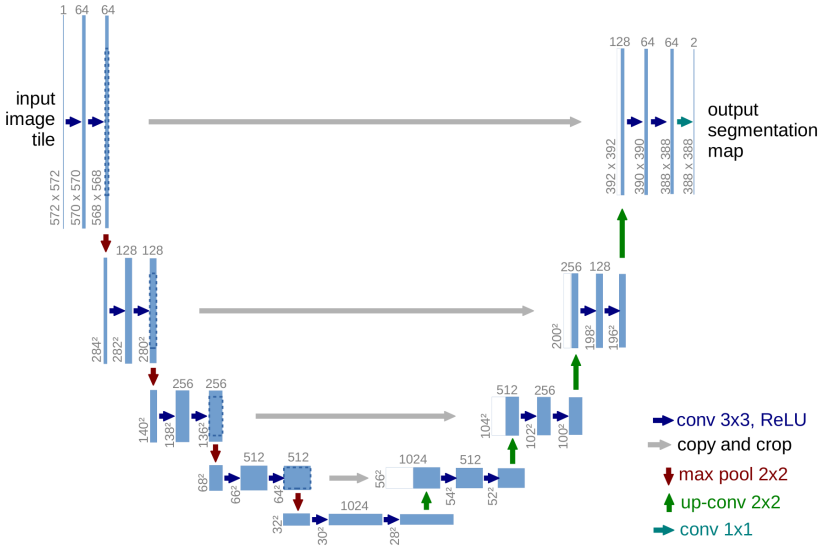
Esse método consiste na utilização de uma série de conexões entre os componentes Encoder e Decoder, como mostra a Figura 2. O objetivo desta técnica é facilitar o processo de decodificação no modelo, conectando os blocos do Encoder diretamente aos blocos correspondentes do Decoder. Esse método não só transfere informação para o Decoder, mas também permite aproveitar o processamento realizado anteriormente e portanto a utilização de camadas menores nesse componente.

Conexões laterais deste tipo são também utilizadas em outras arquiteturas para detecção e segmentação de objetos, como Feature Pyramid Networks (LIN *et al.*, 2016), Pyramid Scene Parsing Network (ZHAO, H. *et al.*, 2016) e Mask R-CNN (HE *et al.*, 2017), com o objetivo de melhorar a performance do modelo em relação à informação espacial. Esse método não só colabora para a construção de máscaras de segmentação mais acuradas, mas também pode ser utilizado para construir redes convolucionais mais robustas às transformações de escala (LIN *et al.*, 2016).

2.4 REDES RESIDUAIS (RESNET)

O avanço na área de visão computacional com o uso de redes convolucionais depende da construção de modelos cada vez mais poderosos. As primeiras demons-

Figura 2 – Arquitetura original utilizada no modelo U-Net (RONNEBERGER; FISCHER; BROX, 2015). À esquerda observamos o componente Encoder, composto por filtros convolucionais e *max-pooling*, e à direita observamos o Decoder, com blocos convolucionais e blocos convolucionais transpostos. As setas em cinza mostram as conexões de atalho entre os dois componentes.



trações de redes convolucionais modernas, como LeNet (LECUN *et al.*, 1998), AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, G. E., 2012) e VGGNet (SIMONYAN; ZISSERMAN, 2015), alcançaram bons resultados a partir do uso de filtros convolucionais em sequência, com um ganho significativo de performance quando utilizado um número maior de camadas.

O aumento no número de camadas em um modelo gera um ganho de performance, porém este ganho é limitado pela capacidade de treinar o modelo de maneira adequada. O treinamento de redes neurais com o uso de *backpropagation* é limitado pelo efeito de desvanecimento do gradiente, efeito que ocorre principalmente em redes que utilizam um número grande de filtros convolucionais e funções de ativação em sequência (GOODFELLOW, I.; BENGIO; COURVILLE, 2016).

O desvanecimento do gradiente é um fenômeno que ocorre durante o treinamento de um modelo quando o gradiente se torna pequeno demais para realizar a otimização dos parâmetros de maneira correta. As funções de ativação utilizadas nas camadas da rede têm a tendência de reduzir o valor do gradiente de maneira cumulativa, de forma que a otimização das primeiras camadas de uma rede com o algoritmo de *backpropagation* tem pouco efeito no valor dos parâmetros.

Esse fenômeno pode ser aliviado com a utilização de diferentes técnicas na construção da rede, como o uso de blocos convolucionais com conexões de atalho. A arquitetura ResNet (HE *et al.*, 2016) permite elaborar redes neurais muito mais profundas a partir da utilização de uma estratégia de conexões residuais entre as camadas. As redes ResNet apresentam uma redução no efeito de desvanecimento

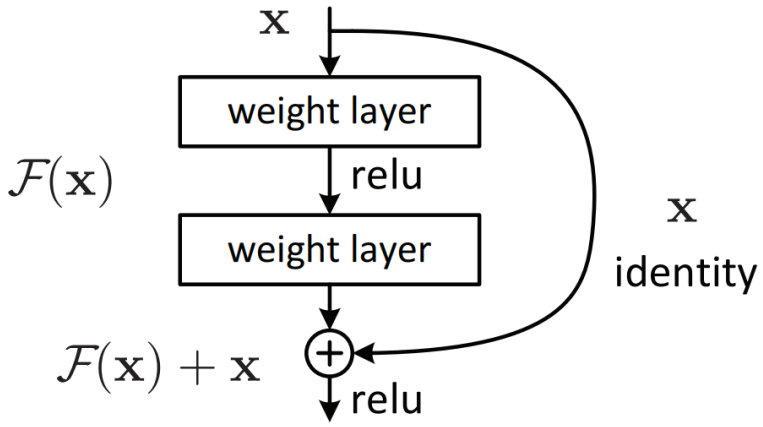
do gradiente e demonstram ganho de desempenho considerável com o aumento da profundidade das redes.

O objetivo dessa arquitetura é permitir a construção de redes neurais mais profundas, com um número significativamente maior de camadas e maior capacidade de aprendizado, sem degradação de desempenho.

2.4.1 Blocos residuais

O bloco residual é o componente primitivo para construção de redes residuais. Essa estrutura é caracterizada pela utilização de conexões de atalho (*shortcut connections*) entre diferentes camadas da rede, o que permite a propagação mais eficiente do gradiente para as camadas iniciais da rede.

Figura 3 – Composição de um bloco residual utilizado na arquitetura ResNet (HE *et al.*, 2016). Cada bloco residual é composto por camadas convolucionais (*weight layer*) e ativações não lineares (relu), que implementam uma função residual $\mathcal{F}(x)$ sobre a entrada do bloco, e uma conexão de atalho entre a entrada e saída do bloco (*identity*). Efetivamente, a função implementada pelo bloco como um todo é a soma da função $\mathcal{F}$ e a saída da conexão residual.



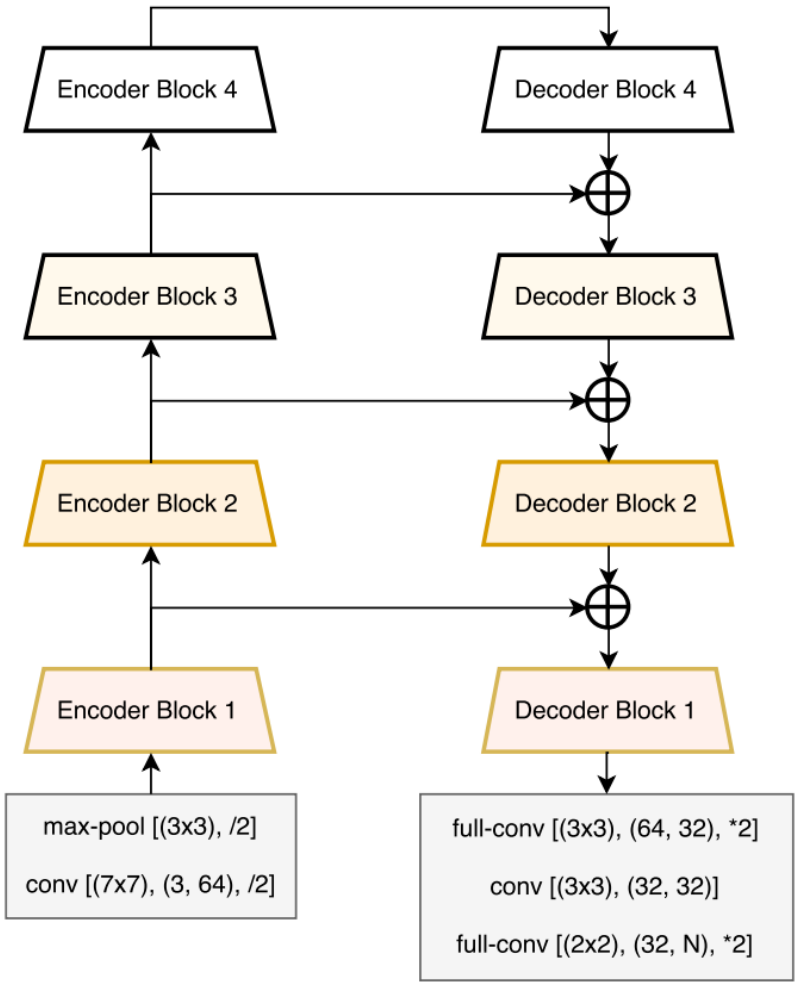
O principal motivo para o sucesso dos blocos residuais está no fato de que a chamada função residual $\mathcal{F}$ realizada pelos filtros convolucionais é apenas a diferença entre os valores de x_{i+1} e x_i, como pode-se observar na Figura 3. Observa-se na prática que a otimização desse tipo de função é mais simples do que a otimização de blocos tradicionais que não utilizam conexões de atalho.

Redes residuais podem então ser construídas a partir da utilização de múltiplos blocos residuais em sequência para obtenção de redes com diferentes profundidades. O uso de blocos residuais permitiu a construção de redes com centenas de camadas sem degradação de desempenho (HE *et al.*, 2016).

2.5 LINKNET

LinkNet (CHAURASIA; CULURCIELLO, 2017) é uma arquitetura puramente convolucional baseada no método Encoder-Decoder para a segmentação semântica de imagens. Como visto na Figura 4, essa rede é inspirada em modelos como U-Net para geração de máscaras de segmentação, utilizando um codificador, decodificador e conexões de atalho.

Figura 4 – Estrutura geral do modelo LinkNet (CHAURASIA; CULURCIELLO, 2017). Composta por dois componentes principais, o Encoder à esquerda e Decoder à direita, essa rede se baseia em outros modelos Encoder-Decoder para realização de segmentação semântica. Cada componente é composto por multiplos blocos, intercalados pelo uso de *skip connections*.



As principais contribuições dessa arquitetura são a utilização de redes residuais no Encoder e a configuração das conexões de atalho junto ao Decoder. Essas modificações permitem a construção de um modelo maior e mais eficiente, bem como a geração de máscaras de segmentação com alto nível de detalhe espacial.

2.5.1 Encoder

A primeira parte do modelo de segmentação é o Encoder, responsável por codificar a entrada da rede para um espaço latente de menor dimensão. A escolha de um Encoder adequado é chave para obter um resultado ótimo para segmentação final, e no caso do LinkNet é utilizado uma rede residual, ResNet, como codificador.

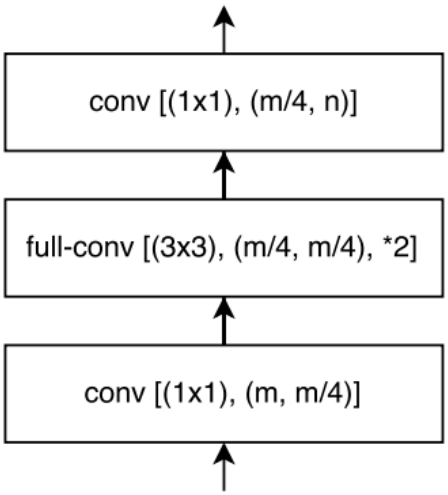
A rede ResNet é utilizada por ser um modelo conhecido pela alta performance e facilidade de treinamento devido à utilização de conexões residuais. Além disso, as conexões residuais da rede trabalham em conjunto com as conexões de atalho dessa arquitetura, realizando uma função similar entre os blocos do Encoder.

2.5.2 Decoder

O Decoder é a segunda parte da arquitetura LinkNet, e é responsável por mapear a representação obtida pelo Encoder em uma máscara de segmentação. O papel do decoder é conseguir produzir uma saída correta, tanto em questão de informação semântica quanto informação espacial, a partir do processamento realizado pelo Encoder.

A arquitetura do Decoder na arquitetura Linknet consiste em blocos compostos por dois filtros convolucionais comuns e um filtro convolucional transposto, sendo este último responsável por realizar a operação oposta ao Max Pooling para recuperar a dimensão original da imagem de entrada.

Figura 5 – Estrutura de um bloco utilizado no componente Decoder do modelo LinkNet (CHAURASIA; CULURCIELLO, 2017). Cada bloco é composto primeiramente por uma camada convolucional, seguida de uma camada utilizando convolução transposta com objetivo de aumentar a resolução do sinal, e finalmente por outra camada convolucional. Os sinais contidos nas conexões de atalho são adicionados à saída de cada bloco contido no Decoder, como ilustrado na Figura 5.



2.5.3 Skip Connections

A utilização de conexões de atalho é essencial para evitar a degradação de desempenho da rede. No modelo LinkNet, os componentes Encoder e Decoder contêm um número igual de blocos de processamento, de forma que as conexões de atalho são feitas entre a entrada dos blocos no codificador para a saída dos blocos correspondentes no decodificador.

No modelo U-Net as conexões de atalho são implementadas pela concatenação dos sinais do codificador com os sinais do decodificador, seguidas de uma camada convolucional. No caso do LinkNet é realizada a conexão de maneira diferente, inspirada no funcionamento das redes residuais. Como ilustrado na Figura 4, o sinal proveniente do codificador é somado ao sinal no bloco correspondente do decodificador, de maneira similar à conexão residual no ResNet.

2.6 DATA AUGMENTATION

Data augmentation é uma técnica utilizada no treinamento de redes neurais que consiste na utilização de variações aleatórias das amostras conhecidas durante o processo de treinamento. O modelo é não somente treinado com as imagens e anotações originais, mas também com um número arbitrário de variações dessas amostras.

O principal objetivo desse método é evitar que o modelo memorize as imagens, e ao invés disso aprenda com as informações relevantes contidas nas amostras. Desta forma, é possível diminuir os efeitos de *overfitting* no treinamento, gerando um modelo que pode ser mais seguramente utilizado em outras imagens.

Data augmentation pode utilizar diferentes transformações sobre as imagens e anotações durante o treinamento, devendo apenas ser escolhida de forma a não modificar as imagens a um ponto em que se perca as informações contidas nela.

Esse tipo de técnica se mostra especialmente efetiva no treinamento com conjuntos reduzidos, onde há um ganho maior na geração de novas imagens e maior probabilidade de observar *overfitting*.

Um dos métodos mais comuns de *data augmentation* consiste na utilização de transformações geométricas nas imagens, como reflexão, rotação, translação, escala e cisalhamento (SHORTEN; KHOSHGOFTAAR, 2019). Para o caso de segmentação semântica, qualquer transformação que altere a geometria da imagem deve ser aplicada também às máscaras para manter a acurácia das anotações.

2.7 DROPOUT

Dropout (KONDA *et al.*, 2015) é uma técnica de regularização utilizada no treinamento de redes neurais com objetivo de reduzir o *overfitting*. Assim como outras

técnicas de regularização, o *dropout* adiciona um custo maior a determinadas configurações dos pesos da rede, estimulando menor interdependência no treinamento do modelo.

Dropout consiste em um método onde um conjunto aleatório de neurônios é ignorado durante o treinamento do modelo a cada iteração do processo. Desta forma, apenas os neurônios mantidos na rede são utilizados na iteração, tanto para predição (*forward*) como na otimização (*backwards*). Existem diferentes abordagens para aplicação de *dropout* em redes convolucionais (SPILSBURY; CAMPS, 2019), e a performance dessa técnica pode variar dependendo do tipo de método e modelo utilizado.

A utilização de *dropout* é similar a adição de ruído à uma rede neural, de forma que a predição feita pelo modelo não depende somente da entrada, mas também de como o *dropout* está estruturado a cada instante. Diferentes iterações sobre uma mesma imagem podem gerar saídas distintas no modelo, o que força o treinamento a gerar um modelo capaz de generalizar melhor a imagem, e ser menos suscetível a sofrer *overfitting*.

2.8 TRANSFER LEARNING E DOMAIN ADAPTATION

Transferência de aprendizado (*transfer learning*) é uma técnica utilizada no treinamento de redes neurais, que consiste na utilização de conjuntos de dados e funções de perda distintos dos de interesse em um processo de pré-treinamento do modelo. O objetivo disso é obter um desempenho superior quando comparado com um modelo inicializado com parâmetros aleatórios (WEISS; KHOSHGOFTAAR; WANG, 2016).

Transferência de aprendizado pode ser utilizada em conjunto com o aprendizado semi-supervisionado, utilizado como uma etapa de pré-treinamento dos modelos. Essa técnica permite a utilização de conjuntos de dados diferentes dos que serão utilizados no modelo final, diminuindo a dependência em amostras do conjunto de interesse.

Como um caso específico de transferência de aprendizado, a adaptação de domínio (*domain adaptation*) é outra técnica que faz uso de conjuntos de dados diferentes dos de interesse para realizar o pré-treinamento do modelo. Essa técnica se distingue da transferência de aprendizado pela restrição imposta à tarefa utilizada, que deve ser a mesma da de interesse (FARAHANI *et al.*, 2020).

2.9 FUNÇÕES DE PERDA E MÉTRICAS DE DESEMPENHO

2.9.1 AUPR - Area Under Precision Recall

A área sob a curva de precisão e sensibilidade (AUPR) é uma métrica bastante útil para analisar a performance de problemas onde há desbalanceamento entre as classes (DAVIS; GOADRICH, 2006). Essa métrica leva em consideração a proporção

de amostras positivas no conjunto, e dá maior importância para classes com poucas amostras do que outras métricas como AUROC (SAITO; REHMSMEIER, 2014). Essa métrica é bastante utilizada na avaliação de algoritmos de segmentação de imagem, onde é comum a predominância de pixels da classe *background*.

Para a segmentação de lesões em imagens médicas é extremamente importante a localização correta das lesões. Como queremos minimizar os casos de falsos negativos, a métrica AUPR se mostra adequada para este tipo de tarefa.

Nas predições feitas por modelos de segmentação, um pixel é considerado positivo se o valor previsto para ele é maior que um determinado limiar de probabilidade, e negativo caso contrário. As métricas de precisão e sensibilidade são calculadas a partir do total de verdadeiros positivos (VP), falsos positivos (FP) e falsos negativos (FN) obtidos a partir de um limiar de probabilidade escolhido.

$$PR = \frac{VP}{VP + FP} \tag{1}$$

$$RE = \frac{VP}{VP + FN} \tag{2}$$

O cálculo da AURP é feito utilizando a curva produzida a partir da variação do limiar de probabilidade no intervalo 0 a 1, resultando em uma métrica única para avaliação da performance dos modelos.

2.9.2 Coeficiente e perda Dice

O coeficiente Sørensen–Dice (SDI) é uma ferramenta estatística utilizada para medir a similaridade entre dois conjuntos, e se tornou popular como métrica de similaridade para validação de máscaras de segmentação (DICE, 1945).

Essa métrica leva em consideração características locais das imagens, como a intersecção entre elas, e questões globais, como a área total das imagens. O coeficiente Dice também é naturalmente robusto a problemas de desbalanceamento de classes, em específico para distinção entre classes de interesse e *background* na imagem.

$$SDI = \frac{2\left|Y \bigcap P\right|}{\left|Y\right| + \left|P\right|} \tag{3}$$

Essa métrica pode ser utilizada como função de perda para a otimização de redes neurais, conhecida como perda Dice (MILLETARI; NAVAB; AHMADI, 2016), e é muito utilizada para o treinamento de modelos de segmentação:

$$L_{Dice} = 1 - \frac{2\sum_{i=1}^{N} y_i p_i + \varepsilon}{\sum_{i=1}^{N} (y_i + p_i) + \varepsilon} \tag{4}$$

onde N é o número de pixels na imagem, p_i e y_i são a predição e anotação para o pixel i, respectivamente, e ε é um termo de suavização para evitar que o denominador seja zero.

2.9.3 Entropia Cruzada

Outra função de perda utilizada para tarefas de segmentação é a Entropia Cruzada (Cross-Entropy) (JADON, 2020). Essa função é avaliada para cada pixel da imagem de maneira independente, retirando-se uma média final sobre todos os pixels.

Essa função de perda sofre principalmente com o desbalanceamento de classes, já que o cálculo do valor final de perda não leva em consideração a frequência de cada classe do problema. Outra variação dessa função consiste na na utilização de Weighted Cross-Entropy, onde é dado um valor distinto de importância para cada classe no problema.

Para o caso específico de Weighted Binary Cross-Entropy (WBCE), temos:

$$L_{CE} = \frac{-1}{N} \sum_{i=1}^{i=N} \alpha.y_i.log(p_i) + (1-\alpha)(1-y_i).log(1-p_i) \tag{5}$$

onde N é o total de pixels em uma imagem, y_i e p_i representam a anotação e predição para o pixel i, respectivamente, e o parâmetro α determina o peso dado à classe $y = 1$ no cálculo da perda.

3 APRENDIZADO SEMI-SUPERVISIONADO

A utilização de aprendizado semi-supervisionado para treinamento de redes neurais é uma área de pesquisa ativa no campo de aprendizado de máquina. Encontra-se aplicações de SSL em diferentes tarefas de aprendizado profundo, como segmentação semântica, detecção de objetos e processamento natural de linguagem.

Existem diferentes requisitos que uma arquitetura de aprendizado semi-supervisionado e conjunto de dados devem seguir de forma que a utilização de amostras não anotadas seja benéfica para o treinamento de um modelo (CHAPELLE; SCHÖLKOPF; ZIEN, 2006). Esses requisitos dizem respeito às características do conjunto de dados e como as amostras estão distribuídas sobre o seu espaço.

Pressuposto de suavidade (*Smoothness assumption*)

Amostras similares em um conjunto devem resultar em predições próximas no espaço de representação, de forma que variações pequenas sobre uma amostra devem resultar em variações pequenas sobre o rótulo. Esse requisito trata sobre o formato da fronteira de decisão de um algoritmo, e reflete sobre a capacidade de generalização e comportamento do modelo sobre amostras de alta dimensionalidade.

Pressuposto de agrupamento (*Cluster assumption*)

Amostras que pertencem a um mesmo grupo tendem a pertencer a uma mesma classe, e amostras de uma mesma classe tendem a formar grupos. Esse requisito trata da localização das amostras dentro do conjunto, e que a distribuição dos dados tende a formar regiões de alta e baixa densidade de amostras. Esse pressuposto não implica necessariamente na existência de um único grupo, mas que classes diferentes não devem pertencer a um mesmo grupo.

Pressuposto de baixa densidade (*Low-density assumption*)

A fronteira de decisão de um conjunto deve passar por regiões de baixa densidade. Em conjunto com o pressuposto de agrupamento, esse requisito pode ser entendido de maneira a preservar os grupos formados no conjunto, caso contrário a região de decisão separaria grupos em diferentes classes. Esse requisito sugere que a utilização de amostras não anotadas pode ser útil para ajustar a região de decisão de um modelo, utilizando-as como maneira de ajustar essa região.

3.1 MÉTODOS NA LITERATURA

A grande variedade de métodos de aprendizado semi-supervisionado surge das diferentes maneiras em que as amostras não anotadas podem ser utilizadas durante o

treinamento do modelo. As principais variações ocorrem na definição das funções de perda e na arquitetura em que as amostras serão inseridas (YANG *et al.*, 2021). Dos diferentes métodos existentes, podemos dividi-los em grupos principais, baseados nas características primárias dos métodos:

1. Métodos de pseudo labeling
2. Métodos generativos (GAN)
3. Métodos de regularização de consistência
4. Métodos híbridos

3.2 MÉTODOS DE PSEUDO LABELING

Os métodos de aprendizado semi-supervisionado com pseudo labels são baseados na geração de anotações para amostras desconhecidas, que subsequentemente são adicionadas no treinamento supervisionado do modelo.

Existem duas abordagens principais dessa classe de métodos, que diferem no número de predições feitas e em como as pseudo labels são geradas. O primeiro tipo é conhecido como *self-training*, onde as predições de alta confiança de um modelo são utilizadas para o treinamento da própria rede. O segundo tipo de método é baseado na discordância entre diferentes predições que são geradas por múltiplos modelos, de forma que um modelo gera pseudo labels para os restantes.

Pseudo-Label (LEE, 2013) é um método bastante simples para treinamento com pseudo labels, onde o modelo é treinado utilizando as amostras anotadas e não anotadas simultaneamente. Para as amostras não anotadas, o treinamento é feito utilizando a classe com maior confiança da rede como sendo a label correta. Essa técnica é equivalente a uma técnica de regularização de entropia, onde a minimização de entropia sobre as amostras não anotadas favorece a geração de regiões de baixa densidade, melhorando a separação das classes do conjunto.

Noisy Student (XIE *et al.*, 2020) realiza o treinamento semi-supervisionado com o uso de um par de modelos, chamados de Teacher e Student, similar ao processo de Knowledge Distillation (KD) (HINTON, G.; VINYALS; DEAN, 2015). Neste método, o modelo Teacher é treinado de maneira supervisionada utilizando as amostras conhecidas, e o modelo Student é treinado de maneira ruidosa, com técnicas como *dropout*, *data augmentation*, *stochastic depth*, utilizando uma combinação das amostras conhecidas e pseudo labels geradas pelo Teacher. Esse processo é repetido múltiplas vezes, utilizando o modelo Student como um novo Teacher, com o objetivo de gerar pseudo labels de maior confiança a cada iteração e permitir a utilização de arquiteturas maiores nos modelos.

SimCLRv2 (CHEN, T. *et al.*, 2020b) é uma versão modificada do algoritmo SimCLR (CHEN, T. *et al.*, 2020a), que realiza o treinamento do modelo utilizando as

amostras não anotadas em duas etapas, tanto em um processo de pré-treinamento não supervisionado quanto em um processo de destilação de informação com o uso de pseudo labels. O treinamento não supervisionado é feito utilizando perda contrastiva no espaço latente do modelo, de forma a maximizar a similaridade entre diferentes representações de uma mesma amostra sobre múltiplas variações aleatórias. A segunda etapa, feita para produzir um modelo menor e melhorar a performance do algoritmo, utiliza as pseudo labels geradas pelo primeiro modelo no treinamento do Student. Esse algoritmo se mostra bastante eficaz, porém o desempenho é altamente dependente no tipo de estrutura utilizada e de um método de treinamento de alta complexidade computacional, utilizando redes e batch-sizes muito grandes com objetivo de otimizar a perda contrastiva e geração de pseudo labels.

O trabalho de Yalniz et al. (YALNIZ *et al.*, 2019) apresenta um método para treinamento semi-supervisionado e weakly-supervised utilizando pseudo labels em um esquema Teacher-Student de treinamento. Esse método é inspirado em destilação de informação e *self-training*, e demonstra um algoritmo robusto para execução de diferentes tarefas. Nessa abordagem a geração de pseudo labels passa por um processo de filtragem, com objetivo de gerar um conjunto de labels de maior confiança. O treinamento do modelo Student é feito inicialmente com o conjunto de pseudo labels em uma etapa de pré-treinamento, e em seguida por um *finetuning* supervisionado utilizando o conjunto de amostras anotadas.

Outra abordagem para utilização de pseudo labels é a utilização de múltiplos modelos com a análise de discordância entre eles, como Tri-Net (CHEN, D.-D. *et al.*, 2018) e Deep Co-Training (QIAO *et al.*, 2018). Nesse tipo de método, é realizado o treinamento em conjunto de múltiplas redes neurais, de forma que a predição de cada uma delas é utilizada para realizar o treinamento das demais.

Co-training (QIAO *et al.*, 2018) é um algoritmo cujo funcionamento depende da existência de duas imagens complementares, geralmente provenientes de diferentes origens, para cada amostra do conjunto. Dois modelos são inicialmente treinados nas imagens conhecidas, gerando um classificador para cada tipo de imagem, e em seguida são geradas pseudo labels sobre o conjunto não conhecido utilizando as predições de maior confiança de cada modelo. O treinamento com as pseudo labels é feito de maneira a minimizar a distância entre as predições feitas pelos dois modelos sobre um par de imagens complementares.

Tri-Net (CHEN, D.-D. *et al.*, 2018) é um método que utiliza as predições geradas por três modelos para geração das pseudo labels. Os modelos são inicialmente treinados simultaneamente, utilizando um módulo compartilhado e um processo conhecido como Output Smearing (BREIMAN, 2000) sobre o conjunto anotado. Em seguida, as pseudo labels são geradas sobre o conjunto não conhecido sempre que dois modelos estão em concordância em relação à predição sobre uma amostra, que é então

adicionada ao conjunto de treinamento do terceiro modelo.

3.3 MÉTODOS GENERATIVOS (GANS)

Arquiteturas GAN (Generative Adversarial Network) (GOODFELLOW, I. J. *et al.*, 2014) são capazes de utilizar imagens não anotadas para o treinamento de redes convolucionais a partir do uso de uma função de discriminação. GANs são compostas por duas redes neurais distintas, chamadas de Gerador e Discriminador, que trabalham em conjunto em um processo de treinamento adversário. A ideia principal dessa arquitetura é modelar a distribuição de dados do conjunto de interesse utilizando o modelo Gerador, e ser capaz de distinguir entre amostras reais e amostras geradas artificialmente utilizando o discriminador.

Uma arquitetura GAN é composta pelo modelo Gerador, capaz de criar amostras artificiais de alta fidelidade em relação ao conjunto de interesse, e pelo modelo Discriminador, capaz de distinguir entre amostras artificiais e amostras reais com alto nível de precisão. O treinamento tradicional de uma GAN é feito de maneira adversarial, onde as imagens criadas pelo modelo Gerador são avaliadas pelo modelo Discriminador, um processo que permite utilizar amostras não anotadas de maneira efetiva, uma vez que é preciso apenas saber se a imagem é artificial ou não para cálculo da função de perda.

O modelo Discriminador treinado em uma GAN pode então ser utilizado como ponto de partida para o treinamento de um classificador. Pode-se também utilizar outras funções de perda junto ao treinamento da GAN com objetivo de gerar um Discriminador com capacidade de classificar as amostras desejadas.

Semi-Supervised GAN (SGAN) (ODENA, 2016) é um modelo que estende o conceito de GAN para permitir o treinamento do Discriminador de maneira mista, utilizando um processo supervisionado e não supervisionado simultaneamente. Nesta arquitetura, o componente Discriminador recebe a tarefa de classificar as amostras recebidas em $K + 1$ classes, onde K representa o número de classes presentes no conjunto anotado, e o termo adicional se refere à distinção entre amostras reais e artificiais.

ImprovedGAN (SALIMANS *et al.*, 2016) e GoodBadGAN (DAI *et al.*, 2017) utilizam também o treinamento com $K+1$ classes para geração de um Discriminador capaz de classificar imagens. Essas arquiteturas contribuem com diferentes técnicas focadas na otimização do treinamento do modelo GAN, e apresentam diferentes funções de perda para o Discriminador com objetivo de corrigir os pontos fracos observados no treinamento misto da SGAN.

Outras tarefas de funções de perda podem ser utilizadas para o treinamento de redes GAN com amostras não anotadas. A arquitetura CCGAN (DENTON; GROSS; FERGUS, 2016), por exemplo, realiza o treinamento dos modelos Gerador e Discrmi-

nador com o uso de *in-painting*. Com esta técnica, o modelo Gerador recebe a tarefa de reconstruir seções de uma imagem a partir da informação contextual restante, e o Discriminador recebe a tarefa de distinguir entre imagens originais e imagens que foram preenchidas artificialmente. O Discriminador treinado nessa arquitetura pode então ser generalizado para um classificador a partir de um processo de *finetuning*.

3.4 MÉTODOS DE REGULARIZAÇÃO DE CONSISTÊNCIA

Os métodos de regularização de consistência são baseados no princípio de continuidade entre as predições de um modelo, seguindo a ideia que amostras similares devem resultar em predições similares (YANG *et al.*, 2021). Desta forma, essa classe de métodos é centrada na ideia de que pequenas perturbações sobre amostras de um conjunto não devem modificar a predição feita por um modelo.

A métrica de regularização de consistência depende da restrição aplicada sobre o projeto a priori, e geralmente se baseia na distância entre múltiplas predições feitas sobre uma amostra. O cálculo de distância entre as predições é geralmente feito utilizando Mean Square Error (MSE) ou KL-divergence.

PI-Model (SAJJADI; JAVANMARDI; TASDIZEN, 2016) utiliza o princípio de suavidade como fundamento para o cálculo de consistência entre predições. Neste método, o próprio modelo realiza duas predições sobre uma mesma amostra utilizando diferentes perturbações aleatórias sobre o sinal, tanto sobre a imagem, na forma da Data Augmentation, quanto no modelo em si, na forma de Dropout e Random Pooling. Por fim, a perda de consistência é calculada diretamente a fim de minimizar a distância entre as duas predições produzidas.

Temporal Ensembling (LAINE; AILA, 2017) é uma arquitetura similar ao PI-Model cuja principal contribuição é a utilização de Exponential Moving Average (EMA) das predições no cálculo da perda de consistência. Diferente do PI-Model, que requer duas predições sob condições diferentes para cada iteração do treinamento, essa técnica utiliza uma média temporal das predições passadas para gerar um valor de referência para o cálculo da perda. Desta forma, o treinamento é realizado calculando-se a perda de consistência entre a predição atual do modelo e a EMA das predições passadas.

Mean-Teacher (TARVAINEN; VALPOLA, 2017) é outra técnica similar ao Pi-Model, que utiliza o conceito de Exponential Moving Average (EMA) com objetivo de melhorar as predições utilizadas no cálculo de regularização de consistência. Essa arquitetura utiliza o EMA sobre os pesos do próprio modelo, ao invés de sobre as predições dele, com objetivo de gerar uma rede mais acurada com o tempo. Este método faz uso de um par de modelos, chamados de Teacher e Student, onde o modelo Student se comporta assim como na abordagem Pi-Model, e o modelo Teacher utiliza a técnica EMA e é composto pela média temporal do modelo Student, $T_t = \alpha T_{t-1} + (1-\alpha)S_t$. Dessa forma, as predições geradas pelo modelo Teacher se comportam de

maneira mais estável, e resultam em uma referência de maior qualidade para o cálculo da perda de regularização de consistência.

3.5 MÉTODOS HÍBRIDOS

Existem trabalhos que fazem uso de diferentes métodos de aprendizado semi-supervisionado no mesmo processo.

FixMatch (SOHN *et al.*, 2020) é um método que utiliza a geração de pseudo labels de forma a impor a regularização de consistência no treinamento do modelo. Isso é feito primeiro gerando-se duas predições para uma mesma amostra, utilizando um algoritmo de *data augmentation* fraco e outro forte. A primeira predição é tomada como *ground-truth* caso haja um alto grau de confiança, e a perda é calculada de forma a enforçar a consistência entre ela e a predição feita com *data augmentation* forte.

Existe uma outra série de algoritmos híbridos que utilizam o conceito de Mixup (ZHANG *et al.*, 2017). Esse é um conceito simples e efetivo, agindo como uma técnica de *data augmentation* que enforça a regularização do modelo. Esse método impõe a restrição de linearidade entre as amostras e predições de um modelo, de forma que combinações lineares entre amostras devem resultar na mesma combinação linear entre as respectivas anotações. Esse método é utilizado em trabalhos como MixMatch (BERTHELOT *et al.*, 2019b) e ReMixMatch (BERTHELOT *et al.*, 2019a), onde amostras não anotadas e pseudo labels são utilizadas junto ao conjunto anotado usando Mixup.

4 METODOLOGIA

4.1 ARQUITETURA TEACHER-STUDENT E PSEUDO LABELS

Utilizamos neste trabalho um método de treinamento semi-supervisionado com uso de pseudo labels e um *pipeline* Teacher-Student.

A escolha desse método se dá por diferentes motivos. O treinamento semi-supervisionado com pseudo labels é uma técnica generalista, e pode ser utilizada em diferentes domínios e tarefas. É possível utilizar métodos encontrados na literatura para classificação, detecção e segmentação de imagens.

A utilização de técnicas como MixMatch e GAN adicionam bastante complexidade no treinamento dos modelos e exigem adaptações para diferentes tarefas de visão computacional.

Outras técnicas dependem de algoritmos com alto custo computacional, dificultando a aplicação prática dos métodos em imagens médicas de maior resolução. Métodos como Noisy Student e Tri-Net são baseados em métodos iterativos e na utilização de múltiplos modelos em conjunto, o que multiplica o custo computacional para o treinamento e geração das pseudo labels. Métodos como SimCLRv2 são bastante promissores, porém são dependentes da utilização de modelos convolucionais grandes e *batch-sizes* significativamente maiores.

Técnicas como Deep Co-training só podem ser aplicadas para problemas muito específicos, pois são dependentes do tipo de domínio utilizado e a disponibilidade de amostras retiradas de múltiplas origens.

A utilização de um esquema Teacher-Student baseado em Knowledge Distillation (HINTON, G.; VINYALS; DEAN, 2015) demonstra bons resultados na geração de um modelo final, e nos dá mais liberdade para escolha das redes convolucionais e técnicas de treinamento. Essa liberdade facilita o processo de otimização de hiperparâmetros do projeto e permite a inclusão de outras técnicas de treinamento sem mudanças na estrutura geral do projeto. Escolheu-se, portanto, utilizar uma abordagem baseada no trabalho de Yalniz et al. (YALNIZ *et al.*, 2019), que demonstra o uso de pseudo labels e um *pipeline* Teacher-Student para classificação de imagens.

4.1.1 Pipeline Teacher-Student

O método de aprendizado semi-supervisionado proposto por Yalniz et al. (YALNIZ *et al.*, 2019) utiliza o paradigma Teacher/Student com pseudo labels na utilização de conjuntos não anotados de larga escala no treinamento de redes neurais. Esse trabalho demonstra resultados do estado da arte no treinamento semi-supervisionado de classificadores em diferentes conjuntos, assim como em tarefas de classificação de vídeos e aprendizado fracamente supervisionado.

Esse método de aprendizado semi-supervisionado é capaz de utilizar as imagens não anotadas no treinamento do modelo em um processo de múltiplas etapas. Fundamentalmente, esse método é independente ao tipo de tarefa a ser executada, e pode ser utilizado em diferentes aplicações como classificação, detecção e segmentação de imagens.

Essa abordagem utilizando o paradigma Teacher/Student é similar ao processo conhecido como Knowledge Distillation (HINTON, G.; VINYALS; DEAN, 2015) (KD), onde o modelo inicial (Teacher) é responsável por aprender uma representação do domínio de interesse, e esse aprendizado é repassado para o modelo Student, geralmente construído com uma arquitetura com número reduzido de parâmetros. O objetivo final deste procedimento é simplesmente obter um modelo Student com performance similar ao Teacher, porém de menor complexidade computacional.

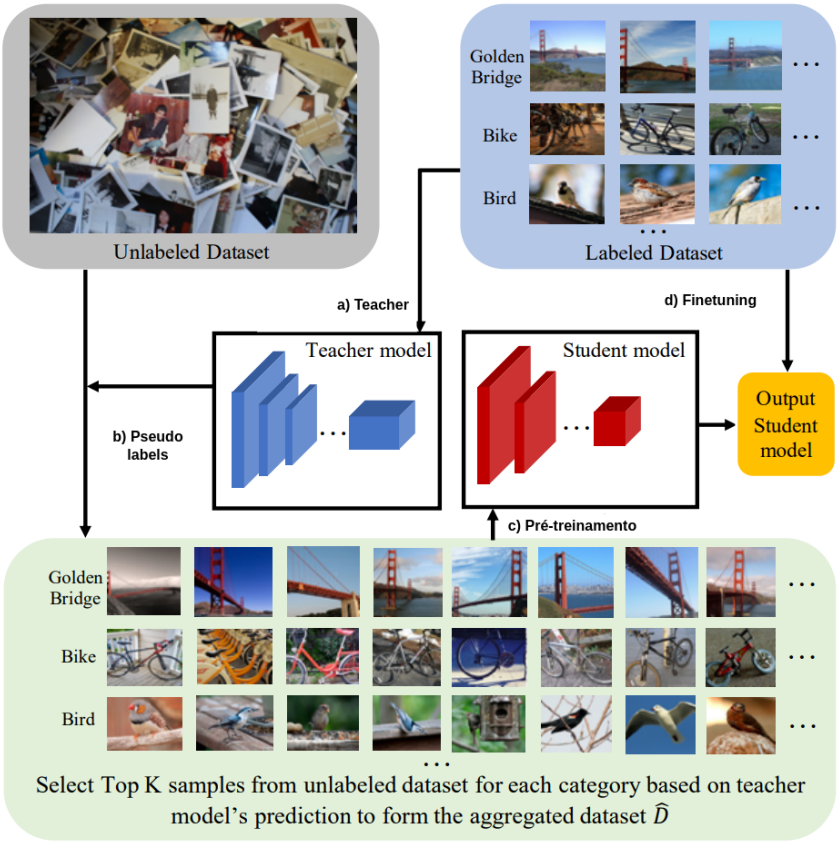
Diferente do processo de Knowledge Distillation, o método Teacher-Student para aprendizado semi-supervisionado se baseia na utilização do modelo Teacher para produção de pseudo labels sobre o domínio não conhecido. O treinamento do modelo Student não é focado na diminuição da quantidade de parâmetros, mas sim na utilização de amostras adicionais no treinamento do modelo a partir do uso de pseudo labels.

O trabalho de Yalniz et al. demonstra a aplicação dessa técnica de aprendizado semi-supervisionado na classificação de imagens, utilizando os *datasets* ImageNet (DENG *et al.*, 2009), YFCC-100M (THOMEE *et al.*, 2015) e IG-1B (YALNIZ *et al.*, 2019). Os resultados obtidos mostram a capacidade do algoritmo de utilizar uma quantidade significativamente maior de amostras não anotadas no treinamento do modelo, e um ganho significativo de desempenho quando comparado com os modelos puramente supervisionados. O algoritmo descrito neste trabalho é focado na classificação multiclasse de imagens, e consiste em quatro etapas fundamentais de treinamento, como visto na Figura 6

A primeira etapa deste método consiste no treinamento supervisionado do modelo Teacher, utilizando o conjunto de amostras anotadas. O objetivo aqui é conseguir um modelo que seja capaz de inferir boas predições sobre o conjunto não anotado, pois as predições feitas sobre esse conjunto serão a base do treinamento do modelo Student.

O modelo Teacher é utilizado para gerar predições, conhecidas como pseudo labels, sobre um segundo conjunto de imagens cuja anotações não são conhecidas. A qualidade das pseudo labels geradas nesta etapa é diretamente dependente da performance do modelo Teacher, e a presença de predições ruidosas ou incorretas é inevitável. Como demonstrado por Yalniz et al, a utilização de pseudo labels de alta qualidade é essencial para obter-se um bom modelo final, e portanto utiliza-se uma etapa de processamento das predições, com objetivo de remover do conjunto as

Figura 6 – Ilustração do processo de treinamento dos modelos Teacher e Student (YALNIZ *et al.*, 2019). Esse processo é feito em quatro etapas, utilizando os dois conjuntos de dados para treinamento do modelo final. **a)** Treinamento supervisionado do modelo Teacher. **b)** Geração de pseudo labels. **c)** Pré-treinamento do modelo Student. **d)** *Fine-tuning* supervisionado do modelo Student.



predições com baixo grau de confiança.

As pseudo labels produzidas pelo modelo Teacher são utilizadas no treinamento do modelo Student, em uma etapa de pré-treinamento utilizando apenas as imagens não anotadas.

Por fim, é realizado o *fine-tuning* do modelo Student, utilizando o conjunto com amostras anotadas para gerar o modelo final.

4.2 CONJUNTO DE TREINAMENTO

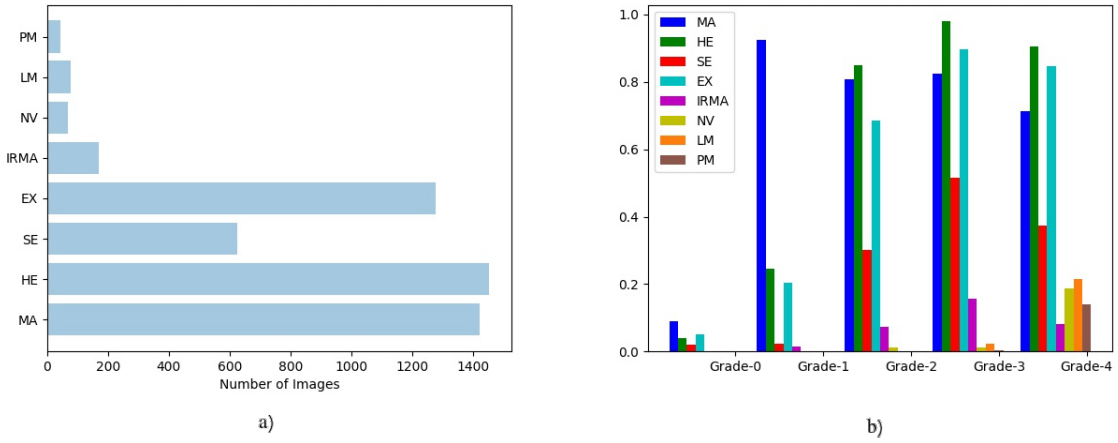
Para realização dos experimentos, escolheu-se utilizar o conjunto de imagens Fine-Grained Annotated Diabetic Retinopathy (FGADR) (ZHOU *et al.*, 2020). Este banco de dados contém imagens utilizadas para o diagnóstico de retinopatia diabética, retiradas na realização de exames de fundo de olho.

No total, o conjunto de dados contém 2.842 imagens de exames de fundo de olho classificadas com o grau de severidade da doença, das quais 1.842 amostras

são segmentadas por oftalmologistas, contendo anotações pixel a pixel dos achados de microaneurismas (MA), hemorragias (HE), exsudatos duros (EX), exsudatos moles (SE), anomalias microvasculares intraretinianas (IRMA) e neovascularização (NV), além da classificação de marcas de laser (LM) e membrana proliferativa (PM) a nível de imagem.

O grau de severidade da retinopatia diabética é avaliado em cinco níveis (0-4): sem presença de RD (0), DR não proliferativa leve (1), RD não proliferativa moderada (2), RD não proliferativa severa (3), e RD proliferativa (4). O nível dado a cada exame depende do número e tamanho de lesões encontradas em cada imagem. A Figura 7 mostra a distribuição de lesões dentro do conjunto e a relação entre o aparecimento de lesões e o nível de severidade da doença.

Figura 7 – Distribuição de lesões encontradas no conjunto FGADR (ZHOU *et al.*, 2020). **a)** Número total de imagens em que cada tipo de lesão é encontrada. **b)** Proporções dos exames em que cada tipo de lesão é encontrada, agrupadas por nível de severidade do exame.

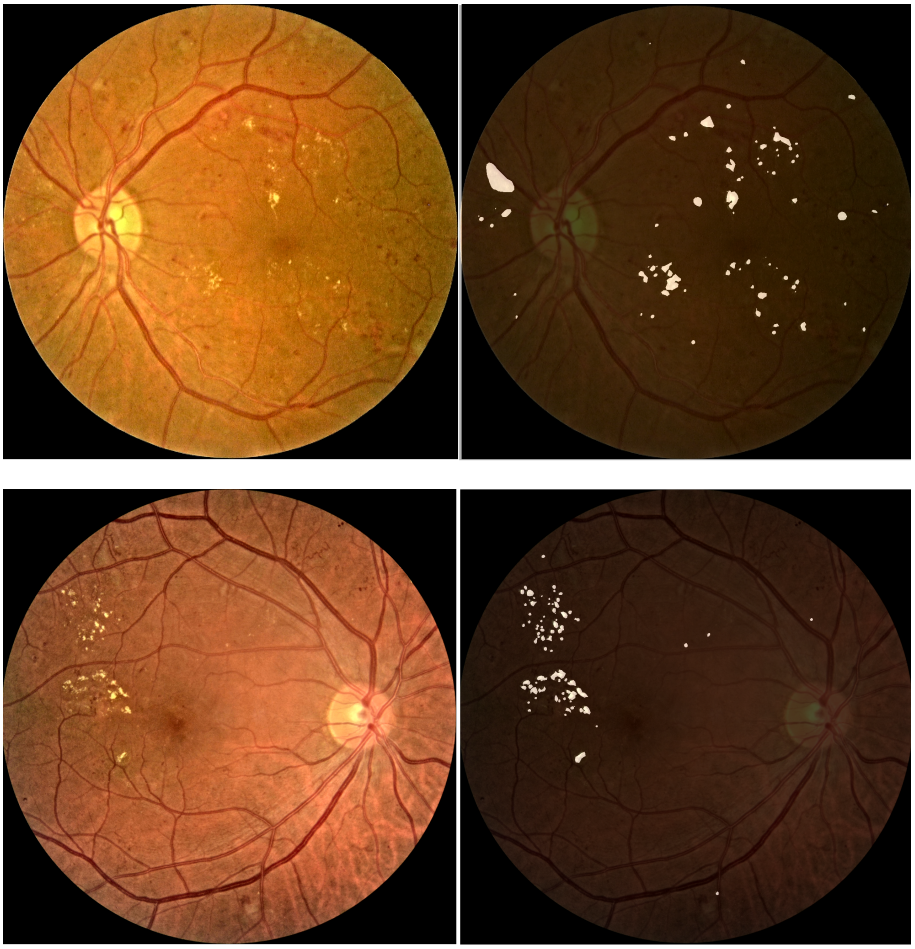


Para realização dos experimentos, será utilizada a classe EX (exsudatos duros) para o treinamento do modelo de segmentação. Essa classe é escolhida para o treinamento por ser um sintoma comum entre os exames, inclusive para casos de baixa severidade, e por demonstrar maior visibilidade nos exames quando comparado com outros sintomas.

A Figura 8 mostra exemplos de exames de fundo de olho com as respectivas anotações relativas à presença de exsudatos duros nos exames. Nota-se que todas as amostras contidas nesse conjunto contém algum tipo de lesão, e é possível que haja algum viés na predição do modelo a partir disso.

Para simular um caso realístico de aprendizado semi-supervisionado, separou-se o conjunto em três partes: um conjunto anotado, utilizado para a parte supervisionada do treinamento; um conjunto não anotado, utilizado para a parte não supervisio-

Figura 8 – Imagens e anotações provenientes do conjunto FGADR. **a)** Imagens de exames de fundo de olho. **b)** Anotações relativas à presença de exsudatos duros, na forma de máscaras binárias.



nada do treinamento; e um conjunto anotado de testes, para realizar a avaliação dos modelos produzidos.

Especificamente, são separadas 16 imagens anotadas para treinamento e 366 imagens anotadas para teste. As 1460 amostras restantes são separadas para geração dos pseudo labels, de forma que ignoramos as anotações existentes para simular um conjunto não anotado de imagens.

4.3 TREINAMENTO DO MODELO TEACHER

A primeira etapa no método Teacher-Student para aprendizado semi-supervisionado é o treinamento do modelo inicial Teacher. Essa etapa consiste no treinamento supervisionado do modelo, utilizando apenas as amostras anotadas, e é essencial para obtenção de um modelo final de boa performance. Esse modelo será utilizado para geração das pseudo labels que serão utilizadas no treinamento do modelo Student, e é importante se ater às restrições do problema em relação ao treinamento supervisio-

nado.

Tendo em vista que a quantidade de imagens anotadas disponíveis é extremamente reduzida, a maior preocupação durante o treinamento do modelo Teacher é com o *overfitting* do modelo sobre o conjunto anotado. A utilização de técnicas para redução de *overfitting* é de extrema importância para o treinamento desse modelo, que deve ser capaz de generalizar o conhecimento adquirido para gerar pseudo labels de qualidade. Para este fim, foram utilizadas as técnicas de *data augmentation*, *dropout* e K-folding durante o treinamento deste modelo.

Para realizar *data augmentation*, foi escolhido uma série de transformações geométricas, de cor e equalização. Cada transformação realizada sobre uma imagem tem um limite de intensidade, escolhido para que a imagem original não seja distorcida demais. As transformações utilizadas foram:

- Escala (entre -20% e 20%)
- Rotação (entre 0 e 180 graus)
- Equalização de histograma CLAHE (com probabilidade de uso 50%)
- Iluminação (entre -20% e 20%, com 20% de probabilidade de uso)
- Contraste (entre -20% e 20%, com 20% de probabilidade de uso)
- Translação (entre -20% e 20%)

Utiliza-se um método de *dropout* conhecido como *Channel Dropout* (TOMPSON *et al.*, 2014) (ou *Spatial Dropout*) aplicado à última camada do modelo. Com essa técnica, alguns filtros da camada escolhida são ignoradas por completo, de maneira aleatória e independente, para cada predição da rede. Esse tipo de técnica de *dropout* é mais efetivo para redes convolucionais, tendo em vista que a remoção de pixels em uma mesma camada é inefetivo quando pixels adjacentes são correlacionados, o que geralmente é o caso em tarefas de segmentação.

O processo de validação de um modelo durante o treinamento supervisionado é essencial para a escolha dos hiperparâmetros do projeto, garantindo boa performance do modelo e acompanhando o ajuste da rede sobre o conjunto de treinamento. O método mais comum para validação de um modelo é a divisão das amostras conhecidas em um conjunto de treinamento e outro de validação. Essa separação de dados é viável quando detém-se um conjunto com amostras em abundância, de forma que a criação de um subconjunto específico para validação não compromete o treinamento do modelo. Neste trabalho, por se tratar de um caso de treinamento semi-supervisionado com um conjunto de amostras anotadas limitado, a validação do conjunto durante o treinamento se torna difícil devido a escassez de dados (OLIVER *et al.*, 2018).

Para contornar este problema, utilizamos a técnica de Validação Cruzada *K-Folding* no treinamento do modelo Teacher, voltado especificamente para realizar a otimização da taxa de aprendizado e *dropout* do modelo.

Para o treinamento do modelo Teacher, fez-se a divisão do conjunto de treinamento em quatro subconjuntos distintos. Cada subconjunto contém 12 imagens que serão utilizadas para treinamento e 4 imagens para validação, geradas de maneira a não repetir as amostras contidas nos conjuntos de validação gerados.

Para otimização dos parâmetros, é realizado o treinamento do modelo Teacher utilizando os quatro subconjuntos de maneira independente, obtendo a média de desempenho dos modelos. O objetivo principal aqui é encontrar valores de taxa de aprendizado e Dropout que gerem um treinamento robusto, com melhor performance e menor variação entre rodadas de treinamento.

Por fim, é realizado o treinamento do modelo Teacher final, utilizando os hiperparâmetros escolhidos com a utilização da Validação Cruzada, com todas as imagens disponíveis e sem realização de validação do modelo.

4.4 PROCESSAMENTO DAS PSEUDO LABELS

Um dos desafios com a geração de pseudo labels utilizando o modelo Teacher está na qualidade das predições feitas por esse modelo. A coleção de predições obtida sobre o conjunto não anotado contém as predições cruas, inclusive aquelas com baixo grau de confiança. A utilização de pseudo labels incorretas ou com baixo grau de confiança gera um conjunto de anotações ruidoso, prejudicando o pré-treinamento do modelo Student e resultando em um modelo final de menor desempenho.

Como mostrado por Yalniz et al. (YALNIZ *et al.*, 2019), utilizar um subconjunto de pseudo labels com maior grau de confiança gera um ganho de desempenho no pré-treinamento do modelo Student quando comparado com a utilização de todas as pseudo labels.

O algoritmo proposto por Yalniz et al. para processamento das pseudo labels consiste na seleção das N melhores predições obtidas para cada classe do problema. Como se trata de um problema de classificação multiclasse em um conjunto desbalanceado, são utilizados os maiores valores obtidos na saída *softmax* da rede, de forma que uma imagem pode ser utilizada para o treinamento de múltiplas classes do conjunto.

Diferente de tarefas de classificação, a função de perda e avaliação do modelo para segmentação de imagens é feito de maneira pixel a pixel, de forma que podemos utilizar um método de processamento que leva essa característica em consideração. Ao invés de remover totalmente uma amostra do conjunto de treinamento, podemos remover apenas os pixels cujas predições tem o menor grau de confiança dentro dessa imagem.

4.4.1 Algoritmo

O algoritmo para filtragem das predições consiste na definição de um intervalo de confiança no qual apenas os pixels que estão dentro desse intervalo serão considerados no cálculo da função de perda. Os limiares de probabilidade são definidos a partir das anotações e predições feitas sobre o conjunto de amostras anotadas.

Dado um conjunto anotado $\mathcal{D}$ com anotações $\mathcal{L}_{\mathcal{D}}$, um conjunto não anotado $\mathcal{U}$:

- Para o conjunto $\mathcal{D}$, calculamos a porcentagem $p_{y=1}$ de pixels da classe $y = 1$, e a porcentagem $p_{y=0}$ de pixels de classe $y = 0$ a partir do conjunto $\mathcal{L}_{\mathcal{D}}$.
- Obtemos o conjunto $\mathcal{P}_{\mathcal{D}}$ de predições sobre o conjunto $\mathcal{D}$.
- Calculamos o limiar de probabilidade T que divide o conjunto $\mathcal{P}_{\mathcal{D}}$ em dois, resultando nas mesmas proporções $p_{y=0}$ e $p_{y=1}$ calculadas anteriormente.
- Obtemos o conjunto $\mathcal{P}_{\mathcal{U}}$ de predições sobre o conjunto não anotado.
- Para cada predição em $\mathcal{P}_{\mathcal{U}}$, calculamos o número de pixels com probabilidade superior a T (N_s), e o número de pixels com probabilidade inferior a T (N_i).
- Para cada predição, mantemos apenas os $K \times N_s$ pixels com maior probabilidade, e os $K \times N_i$ pixels com menor probabilidade. Os pixels fora deste intervalo de confiança são marcados como inconclusivos, e não são utilizados no treinamento do modelo Student.

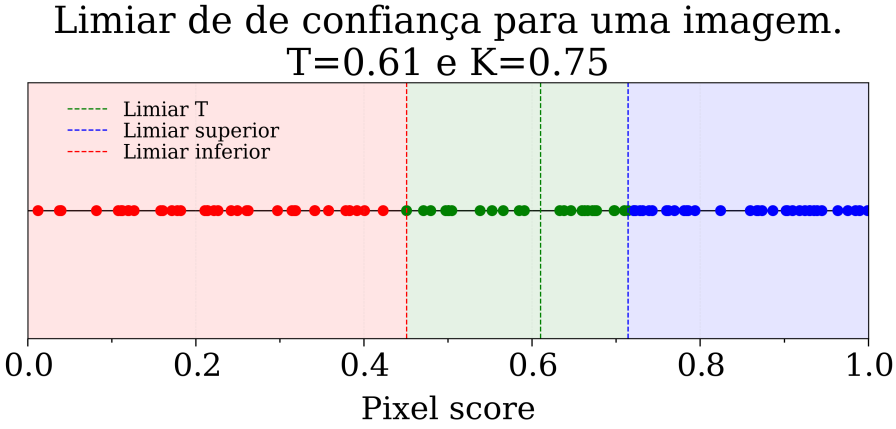
O parâmetro K determina a proporção de pixels que serão mantidos em cada imagem, e pode ser escolhido com fim de realizar um processamento mais ou menos rígido das pseudo labels. O valor utilizado neste trabalho é referente à outros experimentos realizados anteriormente (MARTINS; SILVA, 2021), onde é demonstrado que valores baixos e altos deste parâmetro resultam em perda de desempenho, e o valor ótimo encontrado é $K = 0.75$

Desta forma, como exemplificado na Figura 9, as pseudo labels de cada imagem consistem de três possíveis valores: positivo, negativo e inconclusivo. O treinamento do modelo com essas pseudo labels pode utilizar a mesma função de perda que o treinamento supervisionado comum, modificado apenas para ignorar os pixels inconclusivos do cálculo final.

4.5 TREINAMENTO DO MODELO STUDENT

O treinamento do modelo Student é feito utilizando tanto as imagens anotadas quanto as pseudo labels geradas pelo modelo Teacher anteriormente. Queremos adicionar as imagens não anotadas no treinamento deste modelo de forma a obter uma rede final com performance superior ao modelo Teacher.

Figura 9 – Exemplo do algoritmo de processamento de pseudo labels e a formação dos limiares de confiança para o processamento dos pixels. Supondo um threshold ótimo $T = 0.61$ calculado a partir do conjunto anotado e um parâmetro $K = 0.75$ escolhido, obtemos as seguintes regiões mostradas na figura: em **vermelho** os pixels considerados negativos e usados como pseudo label para classe 0, em **azul** os pixels considerados positivos e usados para classe 1, em **verde** os pixels fora dos intervalos de confiança e desconsiderados no treinamento.



No método apresentado por Yalniz et al, o treinamento do modelo Student é feito em duas partes: uma etapa de pré-treinamento sobre o conjunto $\mathcal{U}$ e uma etapa seguinte de *fine-tuning* supervisionado sobre o conjunto $\mathcal{D}$. Este processo separa completamente o uso dos conjuntos de dados, com as imagens anotadas sendo utilizadas para treinar o modelo Student apenas na etapa final.

Assim como no treinamento supervisionado do modelo Teacher, a utilização de um conjunto tão reduzido de treinamento para o *fine-tuning* do modelo Student pode gerar problemas de *overfitting* ou resultar em pouco ganho de performance. Inspirados na arquitetura Mean Teacher (TARVAINEN; VALPOLA, 2017), vemos que uma alternativa para o processo de treinamento do modelo Student é a utilização de um treinamento misto, utilizando tanto o conjunto anotado quanto as pseudo labels em uma mesma etapa. Isso permite evitar que todas as imagens anotadas sejam utilizadas apenas no final do treinamento, e nos dá liberdade para escolher a maneira em que os conjuntos são combinados.

Neste trabalho, escolheu-se fazer a mesclagem dos conjuntos utilizando um parâmetro P que define a proporção entre amostras conhecidas e pseudo labels utilizadas em cada iteração do treinamento. Para cada *mini-batch*, escolhe-se amostras de $\mathcal{D}$ com probabilidade P e amostras de $\mathcal{U}$ com probabilidade $(1 - P)$, até preencher-se o *mini-batch* por completo.

Nota-se que, assim como no treinamento do modelo Teacher, é necessário realizar a otimização de hiperparâmetros para o modelo Student utilizando validação cruzada. Além do novo hiperparâmetro P a ser escolhido, observa-se que devido a

diferença no número de amostras e épocas utilizadas é possível que haja variação nos valores ótimos de taxa de aprendizado para este modelo.

O modelo Student é por fim treinado utilizando todas as imagens anotadas e com os hiperparâmetros encontrados, sem realizar validação do modelo durante o treinamento. As pseudo labels utilizadas para o treinamento final do modelo Student são aquelas geradas pelo modelo final Teacher.

4.6 ADAPTAÇÃO DE DOMÍNIO

Além do treinamento dos modelos Teacher e Student apenas com as imagens do conjunto FGADR, realizou-se o treinamento dos modelos utilizando adaptação de domínio. Essa técnica é utilizada aqui no treinamento do modelo Teacher com objetivo de produzir um modelo de maior performance e promover a geração de pseudo labels de maior qualidade.

Neste trabalho, avaliamos a utilização de adaptação de domínio em conjunto com aprendizado semi-supervisionado, usando o conjunto de dados Indian Diabetic Retinopathy Image Dataset (IDRiD) (PORWAL *et al.*, 2018) como conjunto de origem.

O conjunto de dados IDRiD contém também amostras anotadas de imagens de exames de fundo de olho utilizados para o diagnóstico de retinopatia diabética. São disponibilizadas um total de 516 imagens classificadas em diferentes níveis de severidade, inclusive exames sem a presença de nenhum sintoma. Desse conjunto, 81 amostras com presença de retinopatia diabética são anotadas por especialistas, produzindo máscaras binárias relativas à presença de hemorragias, microaneurismas, exsudatos duros e exsudatos moles.

Como ambos os conjuntos contém o mesmo tipo de anotação, é possível utilizar as imagens e anotações do IDRID diretamente no treinamento dos modelos. Nesta abordagem, o conjunto de origem é utilizado como uma etapa de pré-treinamento, com subsequentemente refinamento com a utilização do conjunto FGADR.

5 EXPERIMENTOS E RESULTADOS

5.1 DETALHES DE TREINAMENTO

Os experimentos realizados foram feitos utilizando a mesma arquitetura para os modelos Teacher e Student, especificamente a rede Linknet. O encoder utilizado é a rede ResNet34 pré-treinada no conjunto ImageNet. A resolução de entrada dos modelos é de 512x512 pixels, e todas as imagens de treinamento e avaliação foram redimensionadas para este valor.

Todos os modelos foram treinados utilizando o otimizador SGD, com momento 0,9, decaimento de 0,0001 e batch-size de 8. A função de perda utilizada para o treinamento é uma combinação da perda Dice (Equação (3)) e entropia cruzada (Equação (5)) com α = 0,99:

$$L = 10 * L_{CE} + L_{Dice} \tag{6}$$

É utilizada a mesma função de perda para o treinamento dos dois modelos. Para o treinamento do modelo Student utilizando pseudo labels, apenas os pixels selecionados durante a etapa de processamento das pseudo labels serão utilizados para o cálculo da perda, os pixels restantes sendo ignorados da equação.

Todos os experimentos são realizados utilizando uma NVIDIA Tesla P100 16GB via Google Colab, com experimentos implementados em Python utilizando PyTorch e OpenCV.

O tempo médio de treinamento para o modelo Teacher é de 30min, e 4h para o modelo Student.

Todos os modelos são avaliados no mesmo conjunto de teste, contendo 366 imagens anotadas retiradas do dataset FGADR.

5.2 OTIMIZAÇÃO DE HIPERPARÂMETROS

A otimização dos hiperparâmetros para o treinamento do modelo Teacher e Student foi feita utilizando os 4 subconjuntos gerados a partir do conjunto anotado de treinamento. A otimização dos parâmetros é feita de maneira separada para os dois modelos, já que as características de treinamento são distintas para os dois.

5.2.1 Otimização do modelo Teacher

A otimização de hiperparâmetros do modelo Teacher foi feita utilizando 300 épocas de treinamento para cada subconjunto, com a taxa de aprendizado inicial sendo dividida por 10 após 150 épocas.

Como pode se observar nas Tabelas 1 e 2, os valores ótimos encontrados para o treinamento do modelo Teacher foram *LR* = 0,01 e *Dropout* = 0,25. A Figura 10 mostra

as curvas de treinamento para este caso de otimização. O treinamento do modelo final foi feito também com 300 épocas, mantendo a divisão da taxa de aprendizado por 10 após 150 épocas.

	Dropout=0	**Dropout=0,25**	**Dropout=0,5**
Subconjunto 0	**0,501**	0,479	0,457
Subconjunto 1	0,477	0,512	**0,514**
Subconjunto 2	0,464	**0,495**	0,451
Subconjunto 3	0,501	**0,505**	0,480
Média	0,487	**0,497**	0,476

Tabela 1 – Valores de AUPR encontrados para diferentes valores de *dropout* durante a otimização do modelo Teacher, utilizando uma taxa de aprendizado fixa *LR* = 0,1. Os resultados apontam um desempenho superior para o caso *Dropout* = 0,25, com uma AUPR média encontrada na validação cruzada de 0,497.

	LR=5e-1	**LR=1e-1**	**LR=1e-2**	**LR=5e-2**	**LR=5e03**
Subconjunto 0	0,447	0,482	**0,490**	0,480	0,490
Subconjunto 1	0,424	0,496	0,498	**0,499**	0,497
Subconjunto 2	0,482	0,512	**0,516**	0,506	0,516
Subconjunto 3	0,508	0,519	**0,524**	0,513	0,523
Média	0,465	0,502	**0,513**	0,499	0,506

Tabela 2 – Valores de AUPR encontrados para diferentes valores de taxa de aprendizado durante a otimização do modelo Teacher, utilizando a taxa de *Dropout* = 0,25 encontrada na Tabela 1. Os resultados apontam um desempenho superior para o caso *LR* = 0,01, com uma AUPR média encontrada na validação cruzada de 0,513.

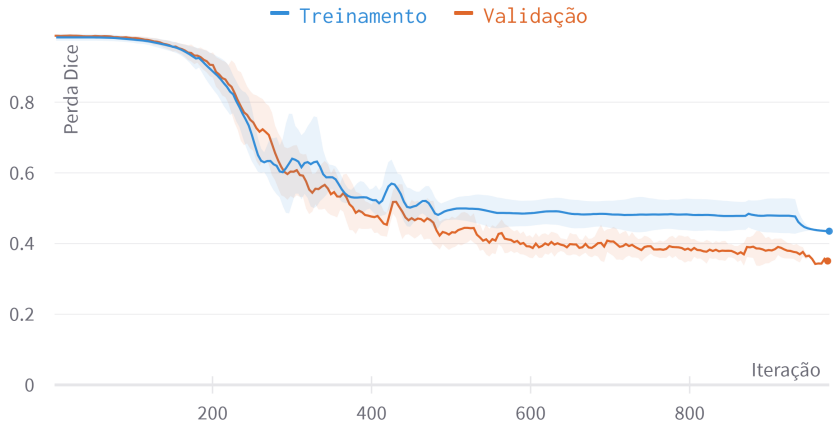
5.2.2 Otimização modelo Teacher com adaptação de domínio

A otimização da taxa de aprendizado para o modelo Teacher pré-treinado com o conjunto IDRID segue o mesmo método utilizado para o modelo Teacher. O modelo Teacher é primeiramente treinado sobre o conjunto IDRID, utilizando todas as 81 amostras disponíveis, e em seguida é feito o *fine-tuning* com o conjunto FGADR. O pré-treinamento do modelo Teacher com o conjunto IDRID é feito durante 200 épocas, com *LR* = 0,01 e *Dropout* = 0,25, mantendo os parâmetros de treinamento restantes iguais.

O processo de *fine-tuning* é feito para cada subconjunto durante 300 épocas, com a taxa de aprendizado sendo dividida por 10 após 150 épocas. Utilizou-se aqui o mesmo valor de *dropout* encontrado na Tabela 1.

Como visto na Tabela 3, o valor ótimo encontrado para este caso é *LR* = 0,005 e *Dropout* = 0,25, e a Figura 11 mostra as curvas de treinamento para este caso

Figura 10 – Curvas de treinamento para o modelo Teacher para otimização do valor de taxa de aprendizado e dropout. No gráfico, mostra-se a perda Dice durante o treinamento e validação do modelo Teacher para o caso LR = 0,01 e $Dropout$ = 0,25. As curvas mostradas no gráfico representam o valor médio da perda Dice para os diferentes subconjuntos, e a área sombreada mostra o desvio padrão entre os treinamentos.



de otimização. Por fim, estes hiperparâmetros são utilizados para treinar o modelo Teacher utilizando todas as imagens do conjunto anotado e adaptação de domínio com o conjunto IDRID. Este modelo é treinado por 300 épocas, dividindo-se a taxa de aprendizado por 10 após 150 épocas.

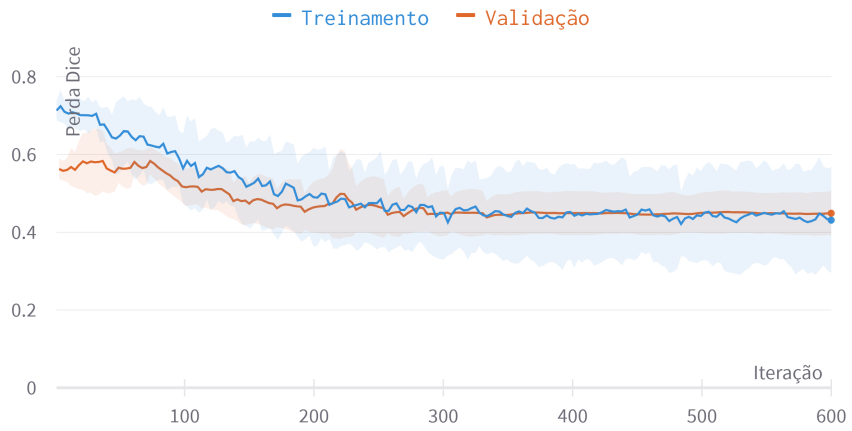
	LR=1e-2	**LR=1e-3**	**LR=5e-3**	**LR=1e-4**
Subconjunto 0	0,518	0,526	**0,530**	0,483
Subconjunto 1	0,524	0,515	**0,530**	0,476
Subconjunto 2	**0,525**	0,502	0,517	0,463
Subconjunto 3	0,537	0,518	**0,539**	0,481
Média	0,526	0,514	**0,529**	0,475

Tabela 3 – Valores de AUPR encontrados para diferentes valores de taxa de aprendizado durante a otimização do modelo Teacher pré-treinado com o conjunto IDRID, utilizando a taxa de $Dropout$ = 0,25. Os resultados apontam um desempenho superior para o caso LR = 0,005, com uma AUPR média encontrada na validação cruzada de 0,529.

5.2.3 Otimização modelo Student

O treinamento do modelo Student é mais complexo do que no modelo Teacher, devido a maior quantidade de imagens utilizadas e pela necessidade de realizar o pré-processamento das pseudo labels antes de realizar o treinamento. Como não observou-se grande variação entre os treinamentos na validação cruzada, escolheu-se realizar a otimização utilizando apenas um subconjunto.

Figura 11 – Curvas de treinamento para o modelo Teacher pré-treinado utilizando o conjunto IDRID para otimização do valor de taxa de aprendizado. No gráfico, mostra-se a perda Dice durante o treinamento e validação do modelo Teacher para o caso LR = 0,005 e *Dropout* = 0,25. As curvas mostradas no gráfico representam o valor médio da perda Dice para os diferentes subconjuntos, e a área sombreada mostra o desvio padrão entre os treinamentos.



Além da taxa de aprendizado, foi analisado também o parâmetro P que define a quantidade de pseudo labels utilizadas no treinamento do modelo Student. Além de valores fixos durante todo o treinamento, foi avaliado a variação do parâmetro durante o treinamento.

O modelo Student é treinado no subconjunto por 50 épocas, com a taxa de aprendizado sendo dividida por 10 após 25 épocas. É utilizado o valor de K = 0.75 e o *dropout* encontrado para o modelo Teacher.

Como visto nas Tabelas 4 e 5, os valores ótimos encontrados para o treinamento do modelo Student foram LR = 0,001, *Dropout* = 0,25 e P = 0,5. O treinamento do modelo final foi feito com 50 épocas, mantendo a divisão da taxa de aprendizado por 10 após 25 épocas.

Tabela 4 – Valores de AUPR encontrados para diferentes valores de taxa de aprendizado durante a otimização do modelo Student, fixando o parâmetro P = 0,5 e com *Dropout* = 0,25. Os resultados apontam um desempenho superior para o caso LR = 0,001, com uma AUPR de 0,588.

	LR=1e-1	**LR=1e-2**	**LR=1e-3**	**LR=1e-4**
Subconjunto 3	0,5718	0,581	**0,588**	0,365

5.2.4 Modelo completamente supervisionado

Para comparar os resultados obtidos com o uso de aprendizado semi-supervisionado, treinou-se um modelo completamente supervisionado utilizando o conjunto

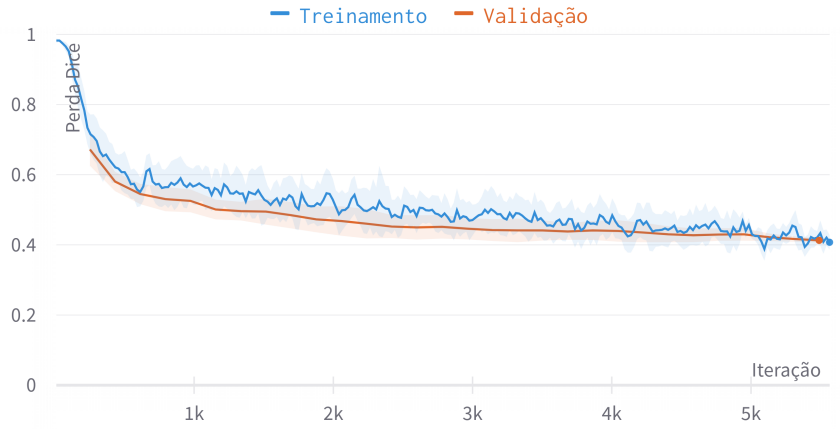
Tabela 5 – Valores de AUPR encontrados para diferentes valores do parâmetro P, utilizando a taxa de aprendizado encontrada na Tabela 4 e com *Dropout* = 0,25. Os casos P = *Cresc* e P = *Desc* consistem na variação linear do parâmetro P entre e 0 e 1 durante as épocas do treinamento, sendo o primeiro caso um variação crescente (iniciando em $P = 0$ e terminando com $P = 1$) e o segundo caso o processo inverso. Os resultados apontam um desempenho superior para o caso $P = 0,5$, com uma AUPR de 0,588.

	P=0,1	**P=0,2**	**P=0,5**	**P=0,8**	**P=0,9**	**P=Desc**	**P=Cresc**
Subconjunto 3	0,559	0,5609	**0,588**	0,5691	0,550	0,551	0,571

FGADR por inteiro durante o treinamento. O objetivo é obter um valor de referência de AUPR obtido com o uso de todas as imagens e anotações, utilizando o mesmo modelo e processo de treinamento.

O treinamento do modelo é realizado utilizando todas as 1.842 imagens anotadas do conjunto FGADR, das quais 1.295 são utilizadas para treinamento, 181 para validação, e mantém-se o mesmo conjunto de testes utilizado para os experimentos anteriores, com 366 imagens. O treinamento é feito utilizando a mesma rede neural, LinkNet com codificador ResNet34, e os detalhes de treinamento seguem os valores especificados na Seção 5.1, incluindo as funções de *data augmentation* utilizadas. O treinamento é feito utilizando a taxa de aprendizado $LR = 0.01$ e *Dropout* = 0.25, e a Figura 12 mostras as curvas de perda para o treinamento do modelo completamente supervisionado.

Figura 12 – Curvas de treinamento para o modelo completamente supervisionado. No gráfico, mostra-se a perda Dice durante o treinamento e validação do modelo utilizando $LR = 0,01$ e *Dropout* = 0,25. As curvas mostradas no gráfico representam o valor médio da perda Dice para diferentes rodadas de treinamento, e a área sombreada mostra o desvio padrão entre elas.



5.3 RESULTADOS

Os experimentos realizados são os a seguir:

1. Treinamento puramente supervisionado utilizando todo o conjunto FGADR para treinamento.
2. Treinamento supervisionado apenas com o conjunto anotado $\mathcal{D}$.
3. Treinamento supervisionado com o conjunto $\mathcal{D}$ e utilizando adaptação de domínio com o conjunto IDRiD.
4. Treinamento semi-supervisionado utilizando os conjuntos $\mathcal{D}$ e $\mathcal{U}$.
5. Treinamento semi-supervisionado utilizando os conjuntos $\mathcal{D}$ e $\mathcal{U}$, e utilizando adaptação de domínio com o conjunto IDRiD.

Os resultados de treinamento são mostrados na Tabela 6. Vemos que o modelo completamente supervisionado obtém uma AUPR maior que os demais, seguido do modelo Student com adaptação de domínio. Observa-se também que a utilização de adaptação de domínio resulta em um ganho de desempenho, tanto para o Teacher quanto Student.

Tabela 6 – Valores de AUPR encontrados para os diferentes experimentos realizados, avaliados sobre o conjunto de teste. Vemos que há um ganho de desempenho com a utilização de pseudo labels no treinamento do modelo Student, com um aumento de 0,513 para 0,587 no treinamento utilizando apenas o conjunto FGADR, e um aumento de 0,542 para 0,594 no experimento utilizando o conjunto IDRiD para pré-treinamento.

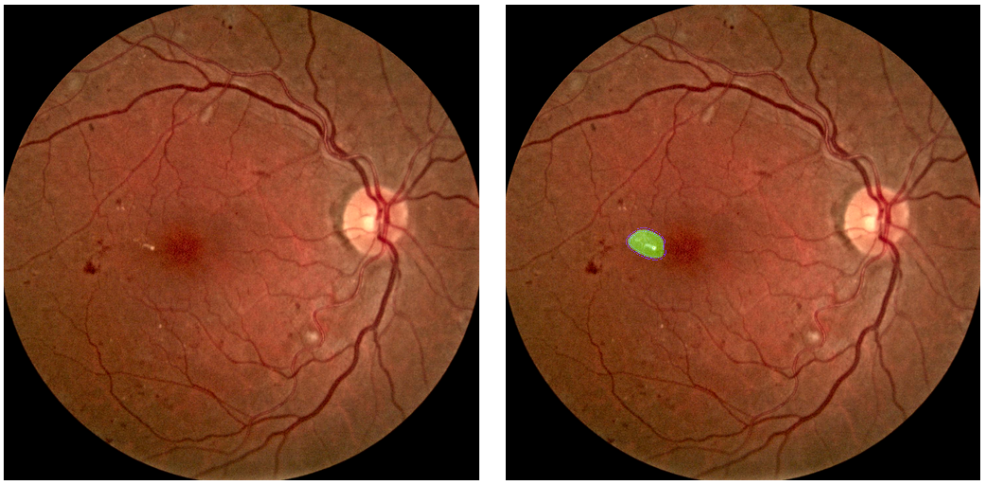
	Run 0	**Run 1**	**Run 2**	**Média**
Completamente Supervisionado	0,659	0,659	0,656	0,665
Teacher	0,519	0,525	0,496	0,513
Teacher com adaptação de domínio	0,542	0,533	0,550	0,542
Student	0,581	0,595	0,584	0,587
Student com adaptação de domínio	0,603	0,576	0,601	0,594

As Figuras 13 a 15 mostram comparações entre as predições feitas pelos diferentes modelos. As imagens originais e anotações são mostradas para comparação, e em seguida as máscaras de segmentação produzidas pelo modelo completamente supervisionado, Teacher pré-treinado com IDRiD e Student pré-treinado com IDRiD. As máscaras de segmentação são coloridas para representar a acurácia dos resultados em relação à anotação: áreas em verde representam os valores de verdadeiro positivo (VP); áreas em azul representam falsos positivos (FP); e áreas em vermelho os falsos negativos (FN).

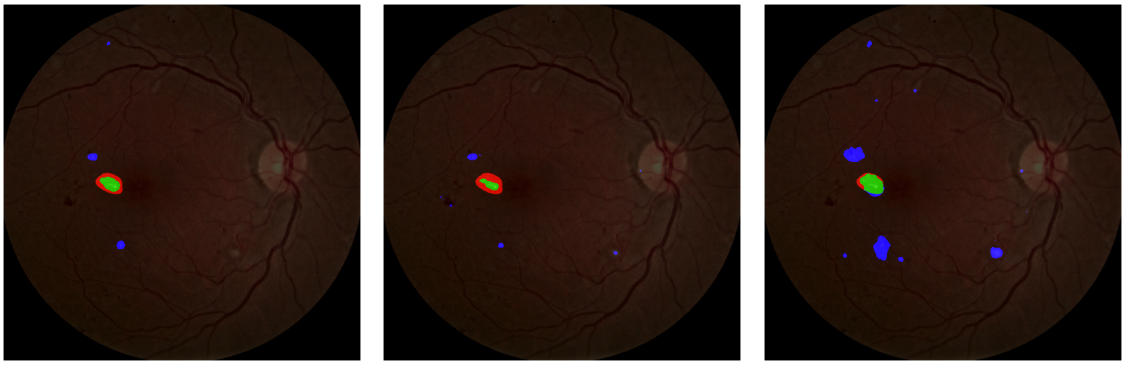
Observa-se que as máscaras geradas pelo modelo Student são mais próximas do modelo completamente supervisionado do que aquelas produzidas pelo modelo Teacher. Em geral o modelo Student obtém uma taxa de verdadeiro positivo próxima a

do modelo completamente supervisionado, porém com uma quantidade maior de falsos positivos. A alta taxa de falsos positivos pode ser observada em todos os modelos, e pode ser influenciada pelas próprias anotações do conjunto, devido ao método de anotação utilizado não ser tão preciso para lesões de menor resolução.

Figura 13 – Predições obtidas na segmentação de uma das imagens do conjunto de teste. Em **a)** temos a imagem original do exame de fundo de olho, com a anotação correspondente em **b)**. As imagens em **c)**, **d)** e **e)** mostram os resultados das predições feitas pelos modelos Supervisionado, Teacher com adaptação de domínio e Student com adaptação de domínio, respectivamente.



(a) Imagem original (b) Anotação

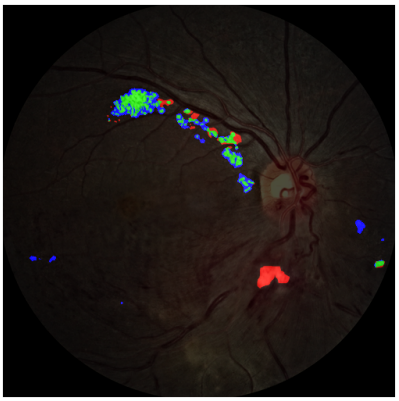


(c) Completamente Supervisionado (d) Teacher (e) Student

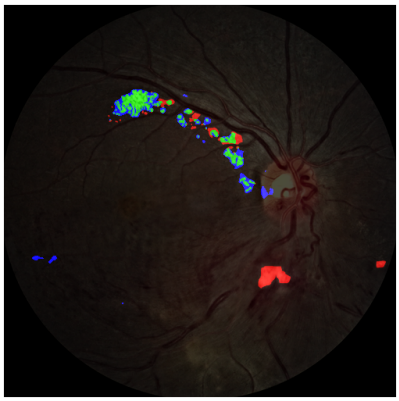
Figura 14 – Predições obtidas na segmentação de uma das imagens do conjunto de teste. Em **a)** temos a imagem original do exame de fundo de olho, com a anotação correspondente em **b)**. As imagens em **c)**, **d)** e **e)** mostram os resultados das predições feitas pelos modelos Supervisionado, Teacher com adaptação de domínio e Student com adaptação de domínio, respectivamente.

(a) Imagem original

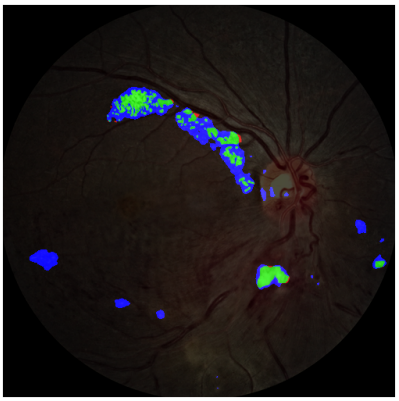
(b) Anotação



(c) Completamente Supervisionado



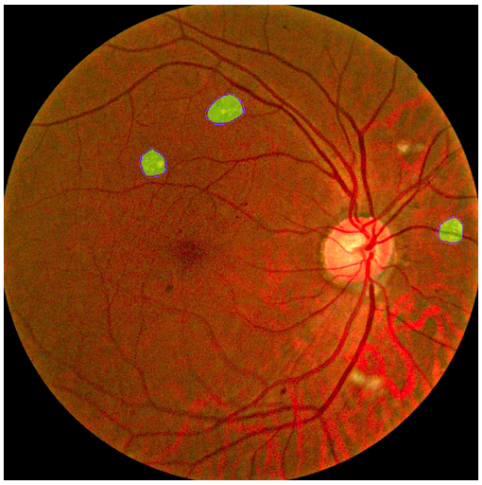
(d) Teacher



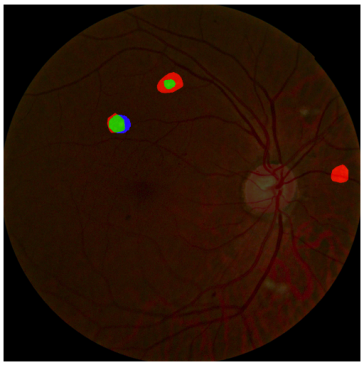
(e) Student

Figura 15 – Predições obtidas na segmentação de uma das imagens do conjunto de teste. Em **a)** temos a imagem original do exame de fundo de olho, com a anotação correspondente em **b)**. As imagens em **c)**, **d)** e **e)** mostram os resultados das predições feitas pelos modelos Supervisionado, Teacher com adaptação de domínio e Student com adaptação de domínio, respectivamente.

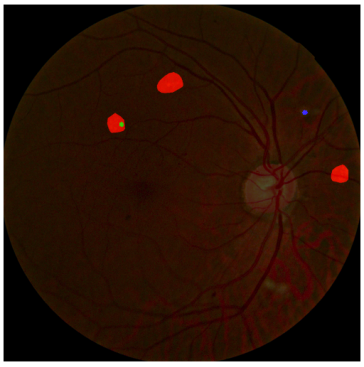
(a) Imagem original



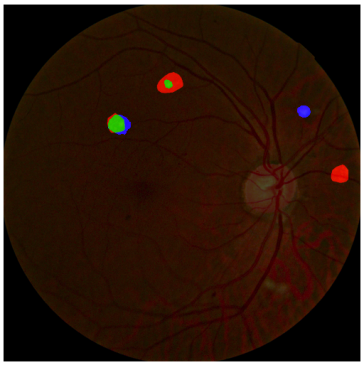
(b) Anotação



(c) Completamente Supervisionado



(d) Teacher



(e) Student

6 CONCLUSÃO

Aprendizado semi-supervisionado é uma técnica que permite o treinamento de redes neurais utilizando amostras não anotadas em conjunto com o treinamento supervisionado. O principal objetivo dessa técnica é permitir o treinamento de modelos utilizando um número reduzido de amostras anotadas, reduzindo a dependência de métodos de aprendizado de máquina em conjuntos de dados grandes.

Teacher-Student é uma abordagem de aprendizado semi-supervisionado que faz uso de um processo de treinamento em múltiplas etapas utilizando pseudo labels. Esse método apresenta bons resultados no desenvolvimento de modelos convolucionais utilizando uma abordagem simples, de baixo custo computacional e que pode ser adaptada para diferentes tarefas de visão computacional.

Neste trabalho analisou-se a utilização de aprendizado semi-supervisionado Teacher-Student para o treinamento de um modelo convolucional aplicado à segmentação de exsudatos duros em imagens de exames de fundo de olho, com objetivo de auxiliar no diagnóstico de retinopatia diabética. Vemos que a utilização deste método resulta em um modelo Student com uma performance significativamente superior à contrapartida simplesmente supervisionada, o Teacher.

Os resultados obtidos mostram que a utilização de aprendizado semi-supervisionado Teacher-Student é uma técnica viável para o treinamento de modelos convolucionais para segmentação semântica em imagens médica. Essa abordagem pode ser utilizada para o desenvolvimento de ferramentas de auxílio ao diagnóstico médico, permitindo maior agilidade e precisão em tarefas como detecção de lesões, análise de imagens histológicas, segmentação de órgãos, entre outras.

6.0.1 Trabalhos futuros

O trabalho apresentado aqui pode servir como base para o desenvolvimento de outras técnicas, permitindo o uso de aprendizado semi-supervisionado em outros domínios e tarefas.

Entre os possíveis trabalhos futuros que podem ser desenvolvidos a partir deste algoritmo, destacamos:

- Aplicação do método de aprendizado semi-supervisionado para segmentação de outros domínios de imagens médicas.
- Adaptação do método para tarefas de segmentação 3D, como segmentação de órgãos em exames de tomografia e ressonância.
- Adaptação do método para problemas tabulares e temporais, como sinais de áudio e eletrocardiograma.

REFERÊNCIAS

ALGAN, Görkem; ULUSOY, Ilkay; GÖNÜL, Şaban; TURGUT, Banu; BAKBAK, Berker. **Deep Learning from Small Amount of Medical Data with Noisy Labels: A Meta-Learning Approach**. [*S.l.*]: arXiv, 2020. Disponível em: https://arxiv.org/abs/2010.06939.

BERTHELOT, David; CARLINI, Nicholas; CUBUK, Ekin D.; KURAKIN, Alex; SOHN, Kihyuk; ZHANG, Han; RAFFEL, Colin. ReMixMatch: Semi-Supervised Learning with Distribution Alignment and Augmentation Anchoring. **CoRR**, abs/1911.09785, 2019. arXiv: 1911.09785.

BERTHELOT, David; CARLINI, Nicholas; GOODFELLOW, Ian J.; PAPERNOT, Nicolas; OLIVER, Avital; RAFFEL, Colin. MixMatch: A Holistic Approach to Semi-Supervised Learning. **CoRR**, abs/1905.02249, 2019. arXiv: 1905.02249.

BREIMAN, Leo. Randomizing Outputs to Increase Prediction Accuracy. **Mach. Learn.**, Kluwer Academic Publishers, USA, v. 40, n. 3, p. 229–242, set. 2000. ISSN 0885-6125.

CHAPELLE, Olivier; SCHÖLKOPF, Bernhard; ZIEN, Alexander. **Semi-Supervised Learning**. [*S.l.*]: The MIT Press, set. 2006. ISBN 9780262255899.

CHAURASIA, Abhishek; CULURCIELLO, Eugenio. LinkNet: Exploiting Encoder Representations for Efficient Semantic Segmentation. **CoRR**, abs/1707.03718, 2017. arXiv: 1707.03718.

CHEN, Dong-Dong; WANG, Wei; GAO, Wei; ZHOU, Zhi-Hua. Tri-net for Semi-Supervised Deep Learning. *In*: PROCEEDINGS of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. [*S.l.*]: International Joint Conferences on Artificial Intelligence Organization, jul. 2018. P. 2014–2020.

CHEN, Liang-Chieh; PAPANDREOU, George; SCHROFF, Florian; ADAM, Hartwig. Rethinking Atrous Convolution for Semantic Image Segmentation. **CoRR**, abs/1706.05587, 2017. arXiv: 1706.05587.

CHEN, Ting; KORNBLITH, Simon; NOROUZI, Mohammad; HINTON, Geoffrey E. A Simple Framework for Contrastive Learning of Visual Representations. **CoRR**, abs/2002.05709, 2020. arXiv: 2002.05709.

CHEN, Ting; KORNBLITH, Simon; SWERSKY, Kevin; NOROUZI, Mohammad; HINTON, Geoffrey E. Big Self-Supervised Models are Strong Semi-Supervised Learners. **CoRR**, abs/2006.10029, 2020. arXiv: 2006.10029.

DAI, Zihang; YANG, Zhilin; YANG, Fan; COHEN, William W.; SALAKHUTDINOV, Ruslan. Good Semi-supervised Learning That Requires a Bad GAN. *In*: GUYON, Isabelle; LUXBURG, Ulrike von; BENGIO, Samy; WALLACH, Hanna M.; FERGUS, Rob; VISHWANATHAN, S. V. N.; GARNETT, Roman (Ed.). **Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA**. [*S.l.*: *s.n.*], 2017. P. 6510–6520.

DAVIS, Jesse; GOADRICH, Mark. The relationship between Precision-Recall and ROC curves. *In*: PROCEEDINGS of the 23rd international conference on Machine learning - ICML '06. [*S.l.*]: ACM Press, 2006.

DENG, Jia; DONG, Wei; SOCHER, Richard; LI, Li-Jia; LI, Kai; FEI-FEI, Li. ImageNet: A large-scale hierarchical image database. *In*: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA. [*S.l.*]: IEEE Computer Society, 2009. P. 248–255.

DENTON, Emily L.; GROSS, Sam; FERGUS, Rob. Semi-Supervised Learning with Context-Conditional Generative Adversarial Networks. **CoRR**, abs/1611.06430, 2016. arXiv: 1611.06430.

DICE, Lee R. Measures of the Amount of Ecologic Association Between Species. **Ecology**, Wiley, v. 26, n. 3, p. 297–302, jul. 1945.

DROZDZAL, Michal; VORONTSOV, Eugene; CHARTRAND, Gabriel; KADOURY, Samuel; PAL, Chris. The Importance of Skip Connections in Biomedical Image Segmentation. **CoRR**, abs/1608.04117, 2016. arXiv: 1608.04117.

FARAHANI, Abolfazl; VOGHOEI, Sahar; RASHEED, Khaled; ARABNIA, Hamid R. A Brief Review of Domain Adaptation. **CoRR**, abs/2010.03978, 2020. arXiv: 2010.03978.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. [*S.l.*]: MIT Press, 2016. (Adaptive computation and machine learning). ISBN 9780262035613.

GOODFELLOW, Ian J.; POUGET-ABADIE, Jean; MIRZA, Mehdi; XU, Bing; WARDE-FARLEY, David; OZAIR, Sherjil; COURVILLE, Aaron C.; BENGIO, Yoshua. Generative Adversarial Nets. *In*: GHAHRAMANI, Zoubin; WELLING, Max; CORTES, Corinna; LAWRENCE, Neil D.; WEINBERGER, Kilian Q. (Ed.). **Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada**. [*S.l.*: *s.n.*], 2014. P. 2672–2680.

GUAN, Hao; LIU, Mingxia. Domain Adaptation for Medical Image Analysis: A Survey. **IEEE Transactions on Biomedical Engineering**, Institute of Electrical e Electronics Engineers (IEEE), v. 69, n. 3, p. 1173–1185, mar. 2022.

HE, Kaiming; GKIOXARI, Georgia; DOLLÁR, Piotr; GIRSHICK, Ross B. Mask R-CNN. **CoRR**, abs/1703.06870, 2017. arXiv: 1703.06870.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep Residual Learning for Image Recognition. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, IEEE, jun. 2016.

HINTON, Geoffrey; VINYALS, Oriol; DEAN, Jeff. **Distilling the Knowledge in a Neural Network**. [*S.l.*: *s.n.*], 2015. arXiv: 1503.02531 [stat.ML].

JADON, Shruti. A survey of loss functions for semantic segmentation. arXiv, 2020.

JÉGOU, Simon; DROZDZAL, Michal; VÁZQUEZ, David; ROMERO, Adriana; BENGIO, Yoshua. The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation. **CoRR**, abs/1611.09326, 2016. arXiv: 1611.09326.

KALPATHY-CRAMER, Jayashree *et al.* **Plus Disease in Retinopathy of Prematurity**. en. v. 123. [*S.l.*]: Elsevier BV, nov. 2016. P. 2345–2351. Disponível em: http://dx.doi.org/10.1016/j.ophtha.2016.07.020.

KONDA, Kishore Reddy; BOUTHILLIER, Xavier; MEMISEVIC, Roland; VINCENT, Pascal. Dropout as data augmentation. **CoRR**, abs/1506.08700, 2015. arXiv: 1506.08700.

KRÄHENBÜHL, Philipp; KOLTUN, Vladlen. Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials. **CoRR**, abs/1210.5644, 2012. arXiv: 1210.5644.

KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. ImageNet Classification with Deep Convolutional Neural Networks. *In*: PEREIRA, F.; BURGES, C. J. C.; BOTTOU, L.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems 25**. [*S.l.*]: Curran Associates, Inc., 2012. P. 1097–1105.

LAINE, Samuli; AILA, Timo. Temporal Ensembling for Semi-Supervised Learning. *In*: 5TH International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. [*S.l.*]: OpenReview.net, 2017.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, nov. 1998. ISSN 0018-9219.

LEE, Dong-Hyun. Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks. **ICML 2013 Workshop : Challenges in Representation Learning (WREPL)**, jul. 2013.

LIN, Tsung-Yi; DOLLÁR, Piotr; GIRSHICK, Ross B.; HE, Kaiming; HARIHARAN, Bharath; BELONGIE, Serge J. Feature Pyramid Networks for Object Detection. **CoRR**, abs/1612.03144, 2016. arXiv: 1612.03144.

LONG, Jonathan; SHELHAMER, Evan; DARRELL, Trevor. Fully Convolutional Networks for Semantic Segmentation. **CoRR**, abs/1411.4038, 2014. arXiv: 1411.4038.

MARTINS, Roberto Augusto Philippi; SILVA, Danilo. On Teacher-Student Semi-Supervised Learning for Chest X-ray Image Classification. *In*: ANAIS do 15. Congresso Brasileiro de Inteligência Computacional. [*S.l.*]: SBIC, jan. 2021.

MEHTA, Sonia. **Retinopatia Diabética - Distúrbios oftalmológicos.** [*S.l.*: *s.n.*], 2020. https://www.msdmanuals.com/pt/profissional/dist%C3%BArbios-oftalmol%C3%B3gicos/doen%C3%A7as-da-retina/retinopatia-diab%C3%A9tica, acesso: 20-06-2019, version:Junho 2020.

MILLETARI, Fausto; NAVAB, Nassir; AHMADI, Seyed-Ahmad. **V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation**. [*S.l.*]: arXiv, 2016. Disponível em: https://arxiv.org/abs/1606.04797.

MONTEIRO, Prof. Doutor Manuel. **Retinopatia diabética - O que é, sintomas.** [*S.l.*: *s.n.*], 2020. https://www.saudebemestar.pt/pt/clinica/oftalmologia/retinopatia-diabetica/, acesso: 29-06-2022, version:01/07/2020.

ODENA, Augustus. Semi-Supervised Learning with Generative Adversarial Networks. **CoRR**, abs/1606.01583, 2016. arXiv: 1606.01583.

OLIVER, Avital; ODENA, Augustus; RAFFEL, Colin; CUBUK, Ekin D.; GOODFELLOW, Ian J. Realistic Evaluation of Deep Semi-Supervised Learning Algorithms. **CoRR**, abs/1804.09170, 2018. arXiv: 1804.09170.

OQUAB, Maxime; BOTTOU, Leon; LAPTEV, Ivan; SIVIC, Josef. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. *In*: 2014 IEEE Conference on Computer Vision and Pattern Recognition. [*S.l.*: *s.n.*], 2014. P. 1717–1724.

PORWAL, Prasanna; PACHADE, Samiksha; KAMBLE, Ravi; KOKARE, Manesh; DESHMUKH, Girish; SAHASRABUDDHE, Vivek; MERIAUDEAU, Fabrice. Indian Diabetic Retinopathy Image Dataset (IDRiD): A Database for Diabetic Retinopathy Screening Research. **Data**, v. 3, n. 3, 2018. ISSN 2306-5729.

QIAO, Siyuan; SHEN, Wei; ZHANG, Zhishuai; WANG, Bo; YUILLE, Alan. Deep Co-Training for Semi-Supervised Image Recognition. **Lecture Notes in Computer Science**, Springer International Publishing, p. 142–159, 2018. ISSN 1611-3349.

RONNEBERGER, Olaf; FISCHER, Philipp; BROX, Thomas. U-Net: Convolutional Networks for Biomedical Image Segmentation. *In*: NAVAB, Nassir; HORNEGGER, Joachim; III, William M. Wells; FRANGI, Alejandro F. (Ed.). **Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III**. [*S.l.*]: Springer, 2015. v. 9351. (Lecture Notes in Computer Science), p. 234–241.

SAITO, Takaya; REHMSMEIER, Marc. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. **PloS one**, v. 10, 2014.

SAJJADI, Mehdi; JAVANMARDI, Mehran; TASDIZEN, Tolga. Regularization With Stochastic Transformations and Perturbations for Deep Semi-Supervised Learning. *In*: LEE, Daniel D.; SUGIYAMA, Masashi; LUXBURG, Ulrike von; GUYON, Isabelle; GARNETT, Roman (Ed.). **Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain**. [*S.l.*: *s.n.*], 2016. P. 1163–1171.

SALIMANS, Tim; GOODFELLOW, Ian; ZAREMBA, Wojciech; CHEUNG, Vicki; RADFORD, Alec; CHEN, Xi. **Improved Techniques for Training GANs**. [*S.l.*]: arXiv, 2016. Disponível em: `https://arxiv.org/abs/1606.03498`.

SALVADOR, Amaia; BELLVER, Miriam; BARADAD, Manel; MARQUÉS, Ferran; TORRES, Jordi; GIRÓ-I-NIETO, Xavier. Recurrent Neural Networks for Semantic Instance Segmentation. **CoRR**, abs/1712.00617, 2017. arXiv: `1712.00617`.

SHORTEN, Connor; KHOSHGOFTAAR, Taghi M. A survey on Image Data Augmentation for Deep Learning. **J. Big Data**, v. 6, p. 60, 2019.

SIMONYAN, Karen; ZISSERMAN, Andrew. Very Deep Convolutional Networks for Large-Scale Image Recognition. *In*: BENGIO, Yoshua; LECUN, Yann (Ed.). **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**. [*S.l.*: *s.n.*], 2015.

SOHN, Kihyuk; BERTHELOT, David; LI, Chun-Liang; ZHANG, Zizhao; CARLINI, Nicholas; CUBUK, Ekin D.; KURAKIN, Alex; ZHANG, Han; RAFFEL, Colin. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. **CoRR**, abs/2001.07685, 2020. arXiv: `2001.07685`.

SPILSBURY, Thomas; CAMPS, Paavo. Don't ignore Dropout in Fully Convolutional Networks. **CoRR**, abs/1908.09162, 2019. arXiv: `1908.09162`.

SULTANA, Farhana; SUFIAN, Abu; DUTTA, Paramartha. Evolution of Image Segmentation using Deep Convolutional Neural Network: A Survey. **Knowledge-Based Systems**, Elsevier BV, v. 201-202, p. 106062, ago. 2020.

TARVAINEN, Antti; VALPOLA, Harri. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *In*: GUYON, Isabelle; LUXBURG, Ulrike von; BENGIO, Samy; WALLACH, Hanna M.; FERGUS, Rob; VISHWANATHAN, S. V. N.; GARNETT, Roman (Ed.). **Advances in**

Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA. [*S.l.*: *s.n.*], 2017. P. 1195–1204.

TEO, Zhen *et al.* Global Prevalence of Diabetic Retinopathy and Projection of Burden through 2045: Systematic Review and Meta-analysis. **Ophthalmology**, v. 128, abr. 2021.

THOMEE, Bart; SHAMMA, David A.; FRIEDLAND, Gerald; ELIZALDE, Benjamin; NI, Karl; POLAND, Douglas; BORTH, Damian; LI, Li-Jia. The New Data and New Challenges in Multimedia Research. **CoRR**, abs/1503.01817, 2015. arXiv: 1503.01817.

TOMPSON, Jonathan; GOROSHIN, Ross; JAIN, Arjun; LECUN, Yann; BREGLER, Christoph. Efficient Object Localization Using Convolutional Networks. **CoRR**, abs/1411.4280, 2014. arXiv: 1411.4280.

WEISS, Karl; KHOSHGOFTAAR, Taghi M.; WANG, DingDing. A survey of transfer learning. **Journal of Big Data**, Springer Science e Business Media LLC, v. 3, n. 1, mai. 2016.

XIE, Qizhe; LUONG, Minh-Thang; HOVY, Eduard; LE, Quoc V. Self-Training With Noisy Student Improves ImageNet Classification. **2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)**, IEEE, jun. 2020.

YALNIZ, I. Zeki; JÉGOU, Hervé; CHEN, Kan; PALURI, Manohar; MAHAJAN, Dhruv. Billion-scale semi-supervised learning for image classification. **CoRR**, abs/1905.00546, 2019. arXiv: 1905.00546.

YANG, Xiangli; SONG, Zixing; KING, Irwin; XU, Zenglin. A Survey on Deep Semi-supervised Learning. **CoRR**, abs/2103.00550, 2021. arXiv: 2103.00550.

ZHANG, Hongyi; CISSÉ, Moustapha; DAUPHIN, Yann N.; LOPEZ-PAZ, David. mixup: Beyond Empirical Risk Minimization. **CoRR**, abs/1710.09412, 2017. arXiv: 1710.09412.

ZHAO, Dan; XU, Guizhi; XU, Zhenghua; LUKASIEWICZ, Thomas; XUE, Minmin; FU, Zhigang. **Deep Learning in Computer-Aided Diagnosis and Treatment of Tumors: A Survey**. [*S.l.*]: arXiv, 2020. Disponível em: https://arxiv.org/abs/2011.00940.

ZHAO, Hengshuang; SHI, Jianping; QI, Xiaojuan; WANG, Xiaogang; JIA, Jiaya. Pyramid Scene Parsing Network. **CoRR**, abs/1612.01105, 2016. arXiv: 1612.01105.

ZHOU, Yi; WANG, Boyang; HUANG, Lei; CUI, Shanshan; SHAO, Ling. A Benchmark for Studying Diabetic Retinopathy: Segmentation, Grading, and Transferability. **CoRR**, abs/2008.09772, 2020. arXiv: 2008.09772.