UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

MICHEL HANZEN SCHEEREN

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA SEGMENTAÇÃO: UM ESTUDO DE CASO EM FOLHAS DE CAFÉ

MEDIANEIRA

2022

MICHEL HANZEN SCHEEREN

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA SEGMENTAÇÃO: UM ESTUDO DE CASO EM FOLHAS DE CAFÉ

Application of deep learning techniques for segmentation: a case study in coffee leaves

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Orientador: Prof. Dr. Arnaldo Candido Junior

Coorientador: Prof. Dr. Pedro Luiz De Paula Filho

MEDIANEIRA

2022

MICHEL HANZEN SCHEEREN

APLICAÇÃO DE TÉCNICAS DE APRENDIZADO PROFUNDO PARA SEGMENTAÇÃO: UM ESTUDO DE CASO EM FOLHAS DE CAFÉ

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Ciência da Computação do Curso de Bacharelado em Ciência da Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 09/junho/2022

Arnaldo Candido Junior
Doutor
Universidade Tecnológica Federal do Paraná - Campus Medianeira

---

Jorge Aikes Junior
Doutor
Universidade Tecnológica Federal do Paraná - Campus Medianeira

---

Nelson Miguel Betzek
Doutor
Universidade Tecnológica Federal do Paraná - Campus Medianeira

MEDIANEIRA

2022

AGRADECIMENTOS

Gostaria de começar expressando toda minha gratidão ao meu pai, minha mãe e meu irmão, por todo o apoio, carinho e suporte financeiro. Não tenho nenhuma dúvida de que não teria chegado até aqui sem eles.

Em segundo lugar, agradeço ao meu orientador Prof. Dr. Arnaldo Candido Junior e ao meu coorientador Prof. Dr. Pedro Luiz De Paula Filho, por toda a ajuda, conselhos e paciência na construção deste trabalho. Agradeço também aos demais professores, por sempre oferecerem o melhor de si para ajudar com o que for preciso.

Quero agradecer também aos amigos que fiz durante toda essa longa jornada acadêmica, em especial ao Darlan, João e Maria. Eles foram de grande importância na superação de todos os obstáculos e dificuldades encontradas ao longo do caminho.

Certamente estes parágrafos não são suficientes para agradecer a todos que fizeram parte dessa importante fase de minha vida. Enfim, meu muito obrigado a todos que contribuíram de alguma forma para a realização deste sonho.

RESUMO

A incidência de doenças sempre foi um problema sério na cultura do café, capaz de causar danos graves e reduzir significativamente a produção e a qualidade do cultivo. Nesse sentido, a identificação precoce e correta dos sintomas causados pelas doenças é uma tarefa importante para permitir tratamento rápido e capaz de mitigar os danos. Uma área da computação capaz de colaborar para amenizar esse problema é a de aprendizado profundo, responsável por melhorar o estado da arte em diversos domínios. No entanto, a utilização de redes neurais profundas para esse propósito ainda enfrenta diversos desafios, relacionados principalmente com a complexidade associada ao uso de imagens em circunstâncias reais, com variações nas condições gerais de captura ou fundos complexos. A aplicação de técnicas de segmentação contribui para uma melhoria significativa dos resultados, já que possibilita a separação dos aspectos relevantes da imagem e permite que a rede se concentre nos elementos certos. Dessa forma, este trabalho propõe a aplicação, avaliação e comparação de técnicas de aprendizado profundo para a segmentação de folhas de café. Foram utilizados cinco modelos de segmentação (U-Net, FPN, DeepLabv3+, CFNet e OCRNet), com quatro diferentes redes extratoras de características (ResNet-50, ResNet-152, DenseNet-121 e EfficientNet-B3). Todos os modelos treinados apresentaram IoU acima dos 83% e *F-score* acima dos 90%, mostrando serem alternativas válidas para a tarefa em questão. Os melhores desempenhos foram obtidos nos treinamentos que combinaram U-Net com DenseNet-121 (87,79% de IoU e 93,23% de *F-score*), U-Net com EfficientNet-B3 (87,10% de IoU e 92,89% de *F-score*), e FPN com DenseNet-121 (87,05% de IoU e 92,82% de *F-score*).

Palavras-chave: aprendizado do computador; redes neurais; cafeicultura.

ABSTRACT

Disease incidence has always been a serious problem in coffee culture, capable of causing severe damage and significantly reducing production. In this sense, the early and correct identification of the symptoms caused by diseases is an important task to allow quick treatment and to mitigate the damage. One area of computing that can help alleviate this problem is deep learning, responsible for improving the state of art in several domains. However, the use of deep neural networks for this purpose still faces several challenges, mainly related to the complexity associated with the use of images in real circumstances, with variations in the general capture conditions or complex backgrounds. The application of segmentation techniques can contribute to a significant improvement of the results, since it enables the separation of the relevant aspects of the image and allows the network to focus on the right elements. Thus, this work proposed the application, evaluation and comparison of deep learning techniques for the segmentation of coffee leaves. Five segmentation models (U-Net, FPN, DeepLabv3+, CFNet and OCRNet) were tested with four different feature extractor networks (ResNet-50, ResNet- 152, DenseNet-121 and EfficientNet-B3). All trained models presented satisfactory performance, with IoU above 83% and F-score above 90%, showing them to be valid alternatives for the task at hand. The best results were obtained in the trainings that combined U-Net with DenseNet-121 (87.79% of IoU and 93.23% of F-score), U-Net with EfficientNet-B3 (87.10% of IoU and 92.89% of F-score), and FPN with DenseNet-121 (87.05% of IoU and 92.82% of F-score).

Keywords: machine learning; neural networks; coffee-growing.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Siglas

CNN	*Convolutional Neural Network*
CPU	*Central Processing Unit*
CVAT	*Computer Vision Annotation Tool*
FPN	*Feature Pyramid Network*
FPS	*Frames per Second*
GPUs	*Graphics Processing Unit*
IoU	*Intersection over Union*
Pascal VOC	*Pascal Visual Object Classes*
PSPNet	*Pyramid Scene Parsing Network*
ReLU	*Rectified Linear Unit*
SGD	*Stochastic Gradient Descent*
SSIM	*Structural Similarity Index*
TPUs	*Tensor Processing Unit*

SUMÁRIO

1 INTRODUÇÃO

Dentre os diversos setores que compõem a base da economia brasileira, a agricultura desempenha papel de suma importância na geração de empregos e renda para o país (OLIVEIRA *et al.*, 2014). Além de ser o terceiro maior produtor de frutas, o Brasil também está entre os maiores produtores agrícolas do mundo, com destaque para as culturas de café, soja, feijão e algodão (ARAGÃO; CONTINI, 2021).

Entre os anos de 2019 e 2020, o Brasil foi o maior exportador de café do mundo, responsável por cerca de 32,2% das exportações globais (ORGANIZAÇÃO INTERNACIONAL DO CAFÉ, 2020). Além da importância econômica, o café também é um forte aliado para a saúde, já que possui atividade antioxidante, promove a redução na taxa de fotoenvelhecimento, tem efeitos anti-inflamatórios, auxilia no tratamento de doenças crônicas, é um estimulante e contribui para melhorias de humor e das atividades cerebrais (CARVALHO *et al.*, 2018).

Um dos fatores mais preocupantes para a produtividade e qualidade do café é a incidência de doenças. Essas apresentam um potencial de causar perdas severas que podem até mesmo inviabilizar a exploração da cultura, um risco tanto para pequenos agricultores de base familiar quanto para os grandes produtores em escala empresarial (FERRÃO *et al.*, 2017). Uma etapa crítica para o controle efetivo da propagação de doenças e que é capaz de minimizar os eventuais danos causados consiste na identificação precoce e precisa de seus sintomas (XIONG *et al.*, 2020).

Apesar da importância, a identificação correta de doenças em plantas é complexa até para a visão humana, apesar de toda a sua notável habilidade natural de reconhecer e interpretar padrões. Por se tratar de uma questão subjetiva, a tarefa é desafiadora e propensa a diversos fenômenos psicológicos e cognitivos, que podem facilmente induzir ao erro. Análises laboratoriais até conseguem identificar doenças com exatidão, mas costumam ser demoradas e apresentar um custo elevado, além de inacessíveis para a maioria dos produtores de pequeno porte (BARBEDO; KOENIGKAN; SANTOS, 2016).

Aplicações que relacionam técnicas de visão computacional com aprendizado profundo tem se destacado na literatura para resolver esse problema. A área de aprendizado profundo também apresentou um crescimento significativo e melhorou drasticamente o estado da arte nos mais diversos domínios (LECUN; BENGIO; HINTON, 2015). Soluções que utilizam métodos de aprendizado profundo demonstram desempenho de ponta quando comparados às abordagens mais tradicionais de aprendizado de máquina em áreas como visão computacional, processamento de imagem, reconhecimento de voz, tradução automática, robótica, cibersegurança, entre outras (ALOM *et al.*, 2019).

No entanto, existem algumas dificuldades relacionadas ao uso de técnicas de aprendizado profundo para a classificação e detecção de doenças em folhas. Em seu artigo, Barbedo (2016) define dois desafios extrínsecos principais relacionados a abordagem: 1) a complexidade presente no fundo da imagem, que pode muitas vezes se assemelhar à própria área de inte-

resse; 2) as condições de captura, tais como variações no brilho, ângulo da captura, distância da foto, o equipamento em que as imagens foram capturadas, entre outros fatores.

Arsenovic *et al.* (2019) demonstraram, a partir de uma série de experimentos realizados, como as redes neurais convolucionais podem atingir precisões de reconhecimento de doenças extremamente altas quando utilizadas com imagens de laboratório, capturadas em condições controladas de fundo e iluminação. No entanto, esse desempenho cai drasticamente quando imagens de campo em condições reais são usadas.

Segundo Barbedo (2019), uma melhora significativa nos resultados de redes neurais convolucionais pode ser obtida a partir da utilização de técnicas de segmentação dos dados. A segmentação gera imagens com características mais homogêneas, contribuindo para a evidenciação dos principais padrões presentes nos dados e permitindo que a rede se concentre apenas nos elementos relevantes. Sharma, Berwal e Ghai (2020) demonstraram em seus experimentos que o uso de imagens segmentadas na etapa de treinamento pode melhorar consideravelmente o desempenho da rede, principalmente quando submetida a dados ainda não apresentados ao modelo.

1.1 Objetivos geral e específicos

O objetivo geral deste trabalho consiste na aplicação, avaliação e comparação de técnicas de aprendizado profundo para a segmentação semântica de folhas de café. Este objetivo pode ser dividido nos seguintes objetivos específicos:

- Selecionar, rotular e expandir bases de imagens da cultura de interesse;
- Selecionar e treinar modelos neurais para a segmentação de folhas de café;
- Avaliar e comparar os resultados obtidos.

1.2 Justificativa

Doenças são um grande problema na agricultura, podendo causar diminuição da produção, prejuízos à qualidade dos produtos e, em casos mais graves, inviabilizar a exploração de determinada cultura (FERRÃO *et al.*, 2017). Nesse contexto, é de suma importância a identificação precoce das enfermidades para permitir um tratamento rápido e eficaz capaz de mitigar os prejuízos causados e proteger a produção (XIONG *et al.*, 2020).

A aplicação de técnicas de aprendizado profundo apresentou crescimento significativo e diversas contribuições relevantes para a área de visão computacional, melhorando consideravelmente o estado da arte em detecção e reconhecimento de objetos (LECUN; BENGIO; HINTON, 2015). No entanto, esses métodos ainda enfrentam diversos problemas relacionados à dificuldade de replicação dos resultados obtidos em ambientes controlados para imagens de

campo em circunstâncias reais, principalmente por conta de variações nas condições gerais de captura da imagem ou fundos muito complexos (ARSENOVIC *et al.*, 2019; BARBEDO, 2016).

Nesse cenário, a aplicação de técnicas de segmentação de imagem pode contribuir com a melhoria dos resultados, pois proporciona a evidenciação dos padrões presentes nas imagens e permite que a rede se concentre nos elementos de interesse (BARBEDO, 2019). Além disso, a tarefa de segmentação também é muito útil em diversas outras aplicações, como no reconhecimento de espécies florestais (PIRES, 2018), estimativa de crescimento de safra (KATAOKA *et al.*, 2003), contagem de folhas (XU *et al.*, 2018), direção autônoma (TREML *et al.*, 2016), identificação de câncer (TREBESCHI *et al.*, 2017), entre outras.

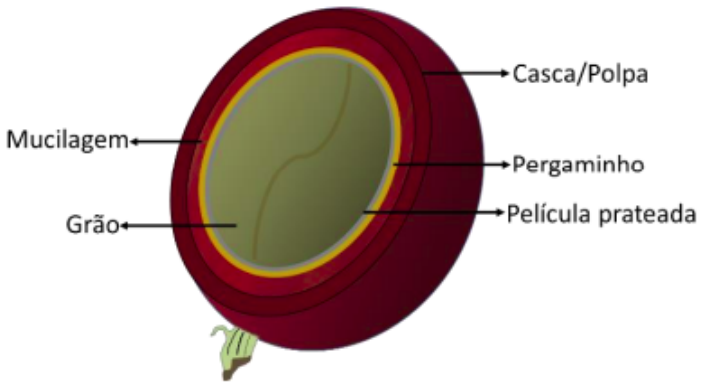
2 REFERENCIAL TEÓRICO

Neste capítulo serão abordados tópicos relacionados à cultura do café (Seção 2.1), tais como suas características e importância econômica. Depois, serão apresentados conceitos referentes a imagens digitais e suas propriedades essenciais (Seção 2.2). Na sequência, serão abordados os assuntos de aprendizado de máquina, redes neurais artificiais e aprendizado profundo (Seções 2.4, 2.5 e 2.6, respectivamente). Em seguida, serão apresentadas noções sobre o processo de segmentação de imagens digitais e dados sintéticos (Seções 2.3 e 2.9). Por fim, as Seções 2.7 e 2.10 trazem modelos de aprendizado profundo e trabalhos relacionados com o tema.

2.1 Café

O cafeeiro é um arbusto da família *Rubiaceae* e do gênero *Coffea*, que se desenvolve principalmente em regiões tropicais e subtropicais e pode atingir de 2 a 2,5 metros de altura. Produz frutos conhecidos como cerejas, que possuem em seu interior dois grãos de café fortemente protegido (MARTINS, 2008). Quatro componentes principais se destacam na estrutura do fruto de café: o exocarpo, também chamado casca; o mesocarpo, conhecido como mucilagem; o endocarpo, também designado de pergaminho; e o endosperma ou grão (DURÁN *et al.*, 2016). A Figura 1 apresenta uma representação do fruto de café e destaca seus componentes principais.

Figura 1 – Fruto de café e seus componentes principais



Fonte: Durán *et al.* (2016).

Segundo Davis *et al.* (2011), já foram catalogadas 124 espécies do gênero *Coffea* em todo o mundo. Dessas, apenas 25 são exploradas comercialmente, embora somente duas sejam realmente relevantes no mercado mundial e responsáveis por quase todo o café consumido no mundo: o *Coffea arabica*, popularmente chamado café arábica; e o *Coffea canephora*, mais conhecido como conilon ou robusta (MARTINS, 2008).

As duas espécies possuem diferenças bem significativas em vários aspectos, como a quantidade de cromossomos, base genética, forma e potencial de produção, capacidade de adaptação ao ambiente, resistência contra pragas e doenças, ciclo de plantio e colheita, além do sabor do fruto e valor econômico (FERRÃO *et al.*, 2017).

2.1.1 Café arábica

É a espécie economicamente mais importante da cultura, responsável pela maior parte do café comercializado mundialmente. Originária da Etiópia, é cultivada em diversas regiões da Ásia, África e América (SOUZA *et al.*, 2004). Apresenta um gosto marcante, de aspecto suave e aromático. É predominantemente aplicada na confecção de cafés especiais, podendo até ser comercializada pura (MARTINS, 2008).

É a única representante tetraploide do gênero *Coffea*, o que significa que possui 44 cromossomos originados a partir de 4 agrupamentos da quantidade de cromossomos básica do gênero. Outra característica importante é o fato de ser autógama, já que a maioria de sua reprodução ocorre por um processo de autofecundação (SOUZA *et al.*, 2004).

Quanto às características físicas, o café arábica apresenta folhas de cor predominantemente verde-escuro e com um leve aspecto brilhante na parte da epiderme superior, além de aspecto ovalado com bordas onduladas. De forma geral, suas folhas possuem cerca de 4 a 6 centímetros de largura por 10 a 15 centímetros de comprimento (SOUZA *et al.*, 2004).

2.1.2 Café conilon

É uma espécie perene originária das florestas baixas da África Equatorial, atualmente cultivada principalmente no sudeste da Ásia, na América do Sul e na África Equatorial (SOUZA *et al.*, 2004). É importante destacar que o café conilon é mais resistente do que o café arábica a fatores abióticos, como deficiências hídricas e nutricionais (FERRÃO *et al.*, 2017).

Outra característica significativa é o fato da espécie ser diploide, o que indica que possui 22 cromossomos a partir de duas cópias do número fundamental do gênero *Coffea*, apenas metade da quantidade presente no *Coffea arabica*. Sua utilização está mais relacionada com a preparação de ligas, normalmente misturadas ao café arábica. Como apresenta um maior rendimento e possui um teor mais concentrado de sólidos solúveis, o conilon é frequentemente usado na criação de cafés solúveis (SOUZA *et al.*, 2004).

Quanto aos aspectos físicos, as folhas do café conilon apresentam forma elíptica e bordas onduladas (SOUZA *et al.*, 2004). Além disso, elas são bem maiores do que as folhas apresentadas pelo *Coffea arabica* e possuem coloração verde bem menos intensa. Possui um porte geral do tipo arbustivo com caule lenhoso (FERRÃO *et al.*, 2017). A Figura 2 exibe a diferença de tamanho e aspecto geral das folhas do café arábica e do café conilon.

Figura 2 – Folhas de café das espécies *Coffea arabica* (esquerda) e *Coffea canephora* (direita)

Fonte: Cocato e D'arc (2020).

2.1.3 Importância econômica

O café é um dos produtos agrícolas de maior importância no mundo, com destaque na geração de emprego e renda para os países produtores e consumidores (PONTE, 2002). Durante a safra dos anos de 2019–2020, a produção mundial de café atingiu a marca de 168,55 milhões de sacas de 60 kg, representando um aumento na produção mundial de cerca de 12,8 milhões de sacas (21,3%) desde a safra dos anos de 2016–2017 (CAMPO & NEGÓCIO, 2021).

Na América do Sul foram produzidas cerca de 78,87 milhões de sacas em 2019–2020, o que significa que o continente foi responsável por 46,8% de toda a produção mundial (CAMPO & NEGÓCIO, 2021). Nesse mesmo período, o Brasil foi o maior exportador de café do mundo, responsável por cerca de 32% de todas as exportações globais (ORGANIZAÇÃO INTERNACIONAL DO CAFÉ, 2020).

Da produção mundial de café do período 2019–2020, 95,73 milhões de sacas eram de café arábica (56,8%) e 72,82 milhões de sacas eram de café conilon (43,2%). No Brasil, foram produzidas 47,37 milhões de sacas do café arábica em uma área de 1,5 milhão de hectares, e 14,25 milhões de sacas em 369,6 mil hectares de café conilon (CAMPO & NEGÓCIO, 2021).

2.1.4 Distúrbios do café

O desequilíbrio no ambiente agrícola é facilmente capaz de induzir a mudanças nas relações entre seres vivos e seus inimigos naturais. O processo causa o surgimento de altas populações de fungos, bactérias, vírus, insetos e plantas. Esses competem com as plantas de interesse econômico da área por água, nutrientes, espaço e luz, ou ainda atacam seus componentes vegetais (FERRÃO *et al.*, 2017).

Diversos distúrbios podem afetar o desenvolvimento do café, causando problemas que podem levar a diminuições drásticas de produção, declínio da qualidade e, em casos graves, à inviabilização da exploração econômica da lavoura (MESQUITA *et al.*, 2016). Assim, controlar a

incidência desses distúrbios é uma tarefa indispensável do ponto de vista econômico, seja por procedimentos químicos, genéticos ou culturais (CARVALHO; CHALFOUN; CUNHA, 2013).

São exemplos de doenças do café de grande impacto econômico e causadores de prejuízos às lavouras: a ferrugem, causada pelo fungo *Hemileia vastatrix Berk. et Br.*; a cercosporiose, provocada pelo fungo *Cercospora coffeicola Berk & Cook*; e a phoma, que tem como agente causador o fungo *Phoma costarricensis*. Diversas pragas também apresentam perigo à cultura do café, destacando-se: o bicho-mineiro; relacionado a mariposa *Leucoptera coffeella*; e o ácaro vermelho, ligado ao ácaro *Oligonychus ilicis* (MESQUITA *et al.*, 2016).

2.2 Imagem digital

Matematicamente, uma imagem pode ser definida como uma função de duas dimensões $f(x, y)$, em que x e y descrevem as coordenadas espaciais e f equivale ao brilho ou intensidade da imagem naquele ponto específico. Quando x, y e f são valores finitos e discretos, estes representam uma imagem digital (GONZALEZ; WOODS, 2018). Para Solomon e Breckon (2011), uma imagem digital pode ser descrita como uma representação discreta de dados com informações espaciais e de intensidade, ou seja, dados do layout e cor de uma cena. Por convenção, a origem de uma imagem sempre está localizada no canto superior esquerdo.

Em termos físicos, pode-se representar uma imagem como o produto de dois componentes: a luz que incide na cena (iluminância); e a luz refletida pelos objetos na cena (reflectância). O valor da iluminância é determinado pela fonte de luz, enquanto a reflectância é definida pelas características dos elementos na cena (PEDRINI; SCHWARTZ, 2007). Portanto, uma imagem digital bidimensional $I(m, n)$ retrata a resposta obtida por um sensor para uma série de posições fixas (m = 0, 1, 2... M; n = 0, 1, 2... N) representadas em um sistema de coordenadas cartesianas de duas dimensões. Cada elemento individual da imagem $I(m, n)$, em que m indica a linha e n a coluna, é chamado *pixel* (SOLOMON; BRECKON, 2011).

2.2.1 Digitalização

A obtenção de uma imagem digital a partir de valores contínuos é feito por um processo chamado digitalização, que pode ser subdividido em dois processos: amostragem e quantização. A amostragem se baseia na definição do domínio da imagem nas direções x e y, gerando uma matriz bidimensional de $M \times N$ amostras. Já a quantização consiste na escolha do valor inteiro que melhor descreve a intensidade para cada um dos *pixels* da imagem (PEDRINI; SCHWARTZ, 2007).

A quantização gera um valor em uma faixa de valores bem definida chamada escala de cinza. A intensidade de uma imagem monocromática em um determinado ponto é chamada de nível de cinza da imagem naquele ponto. Uma representação visual muito comum e amplamente

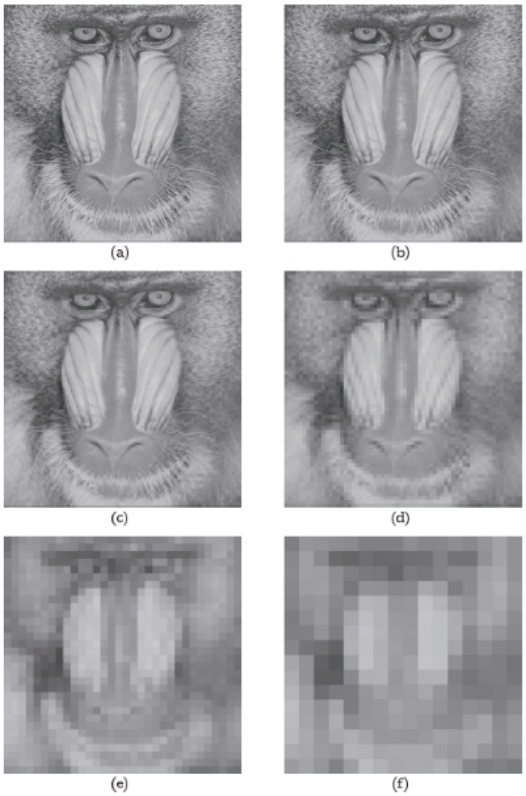
utilizada dessa abordagem é atribuir preto ao nível de cinza mais baixo e branco ao nível de cinza mais alto, ou seja, 0 e 255, respectivamente (PEDRINI; SCHWARTZ, 2007).

Outra representação frequentemente relacionada com a cor dos *pixels* de uma imagem é feita por um vetor triplo, em que cada posição representa a intensidade das cores primárias vermelho, verde e azul. Nesse caso, a imagem é uma combinação linear dos três canais de cores e também pode ser representada por três planos bidimensionais distintos (SOLOMON; BRECKON, 2011).

2.2.2 Resolução

A resolução de uma fonte de imagem pode ser definida de várias formas. A mais comum delas é a resolução espacial, que está fortemente relacionada com a densidade de *pixels* presente na imagem (PEDRINI; SCHWARTZ, 2007). Neste modelo, a quantidade de colunas (C) e linhas (L) da imagem define a quantidade de *pixels* usados para preencher o espaço. Normalmente, essa resolução é expressa como $C \times L$: 640×480, 1280×720, 1920×1080 (SOLOMON; BRECKON, 2011). A Figura 3 ilustra o efeito da redução da resolução espacial.

Figura 3 – Efeitos da redução da resolução espacial (em *pixels*). (a) 512×512**; (b)** 256×256**; (c)** 128×128**; (d)** 64×64**; (e)** 32×32**; (f)** 16×16



Fonte: Pedrini e Schwartz (2007).

Outra forma comum é a resolução temporal, mais relacionada com um sistema de captura de vídeos. Aqui, a resolução é expressa por meio da quantidade de imagens capturadas a

cada intervalo fixo de tempo, como no caso da unidade *Frames per Second* (FPS). Televisões, por exemplo, normalmente operam na casa dos 25 FPS (SOLOMON; BRECKON, 2011).

Por fim, pode-se representar a resolução de uma imagem a partir da quantidade de níveis de cores que cada *pixel* pode assumir. Por exemplo, os *pixels* de uma imagem binária podem assumir apenas as cores preto ou branco, necessitando de apenas um *bit* para sua representação. Já uma imagem em escala de cinza pode variar entre 256 níveis de cinza, necessitando de 8 *bits* para ser representado (SOLOMON; BRECKON, 2011).

2.3 Segmentação

Sob uma perspectiva mais generalista, a segmentação pode ser definida como uma técnica de processamento de dados utilizada para separar uma imagem digital em duas ou mais regiões de interesse. Outra forma de abordar o processo é pensando nele como uma etapa de definição de limites entre entidades semanticamente diferentes, mas presentes em uma mesma imagem digital (GHOSH *et al.*, 2019).

De um ponto de vista mais técnico, a segmentação consiste no processo de atribuição de um rótulo a cada *pixel* de uma imagem digital. Pode-se pensar no método como uma tarefa de classificação em que, em vez de atribuir um único rótulo a toda a imagem, a atribuição é feita a cada pixel individualmente (VASILEV *et al.*, 2019). Atienza (2020) complementa, afirmando que a segmentação particiona uma imagem em um conjunto de regiões com visando entender melhor o que está sendo representado.

Segundo Garcia-Garcia *et al.* (2017), a segmentação pode ser considerada um dos problemas de maior relevância da área de visão computacional. De forma geral, o procedimento consegue proporcionar uma maior compreensão dos objetos e suas relações em uma cena, tarefa importante considerando a quantidade de aplicações de visão computacional ligadas a extração de conhecimento a partir de imagens digitais.

O campo da segmentação de imagens inclui uma grande variedade de problemas e aplicações. Certamente, a versão mais comum e utilizada é a segmentação semântica. Ela consiste na classificação de cada *pixel* da imagem em uma classe, de forma que todos em um mesmo grupo estejam semanticamente relacionados na cena. Vale destacar que o processo pode não depender unicamente dos dados, mas também do problema que se deseja resolver (GHOSH *et al.*, 2019). A Figura 4 apresenta um exemplo de segmentação semântica.

Figura 4 – Exemplo do processo de segmentação semântica de uma imagem digital

Fonte: Adaptado de Ouaknine (2019).

Outra aplicação comum é a segmentação ao nível de instância. Nesse caso, o interesse está em cada objeto contável de uma cena de forma independente. Essa abordagem é muito útil, por exemplo, em aplicações de navegação autônoma, dado que cada componente do trânsito (pedestres, carros, placas de sinalização, cruzamentos...) precisa ser identificado individualmente para poder ser interpretado da forma correta (ATIENZA, 2020). A Figura 5 apresenta um exemplo de segmentação de instância.

Figura 5 – Exemplo de segmentação de instância

Fonte: Ouaknine (2019).

Ghosh *et al.* (2019) também destacam algumas outras tarefas relevantes da área de segmentação, como: identificar e destacar um único objeto de interesse especial na cena, como no caso da chamada detecção de saliências; separar o primeiro plano e o fundo, comum quando se busca minimizar interferências externas na análise de objetos específicos de uma imagem; e rastrear objetos enquanto se movimentam pela cena, como em sistemas de monitoramento de trânsito ou de rastreamento de veículos.

2.4 Aprendizado de máquina

O aprendizado de máquina, também conhecido como *machine learning*, é uma área da computação dedicada a construção de programas de computador que podem aprender e melhorar automaticamente através da experiência. Os conceitos principais do campo de aprendizado de máquina são o resultado do aprimoramento de diversos domínios, incluindo a estatística, inteligência artificial, complexidade computacional, teoria de controle, teoria da informação, ciência cognitiva, biologia e até filosofia (MITCHELL, 1997).

É uma ferramenta computacional bastante valiosa para a resolução de problemas que seriam muito complexos de serem resolvidos por programas convencionais criados por seres humanos. No entanto, para que um bom resultado seja alcançado, é de suma importância a utilização de um conjunto de dados capaz de fornecer todas as informações necessárias para que o algoritmo tenha um aprendizado satisfatório do problema (GOODFELLOW; BENGIO; COURVILLE, 2016). Pode-se perceber uma quantidade crescente de aplicações dessa tecnologia, seja na detecção automática de spam, marcações de fotos em redes sociais, sistemas de reco-

mendação de conteúdo, carros autônomos, dentre muitas outras (DATA SCIENCE ACADEMY, 2021).

Existem três abordagens principais no campo do aprendizado de máquina: aprendizado supervisionado, não supervisionado e por reforço. No aprendizado supervisionado são utilizados dados com seus respectivos resultados esperados, ficando a cargo do algoritmo descobrir como recriar as saídas a partir da entrada. Já no aprendizado não supervisionado nenhum valor esperado de saída é fornecido, ou seja, o algoritmo é totalmente responsável por encontrar e descrever os padrões presentes nos dados. Por fim, o aprendizado por reforço se baseia na ideia de recompensa ou punição para o algoritmo conforme ele progride no treinamento e tenta aprender com os dados (VASILEV *et al.*, 2019).

Segundo Goodfellow, Bengio e Courville (2016), muitas tarefas do dia a dia podem ser desempenhadas por algoritmos de aprendizado de máquina, tais como:

- Classificação: categorizar a entrada de acordo com grupos pré-definidos, normalmente aplicado a tarefas de classificação de imagens;
- Regressão: prever um valor numérico a partir de um conjunto de dados de entrada, muito utilizado no ambiente financeiro para prever o preço de ações e imóveis;
- Transcrição: converter um conjunto não estruturado de dados em um modelo estruturado, como no caso de reconhecimento de *caracteres* a partir de imagens;
- Tradução: reescrever informações em diferentes idiomas, o que é principalmente aplicado à linguagem natural;
- Detecção de anomalias: encontrar inconsistências ou pontos incomuns e fora dos padrões do conjunto de dados.

Apesar do todo o potencial das técnicas de aprendizado de máquina, existem dois desafios intrínsecos relacionados à área: o *underfitting* e o *overfitting*. O *underfitting* ocorre quando o algoritmo não consegue alcançar um desempenho satisfatório durante o treinamento, ou seja, não consegue aprender bem com os dados (GOODFELLOW; BENGIO; COURVILLE, 2016). Já o *overfitting* acontece quando o algoritmo consegue um bom desempenho durante o treinamento, mas se sai mal quando avaliado com dados novos, estando normalmente relacionado com a memorização do conjunto de treinamento (TAYLOR, 2017).

2.5 Redes neurais artificiais

Uma rede neural artificial é um modelo computacional inspirado na maneira como o cérebro humano funciona. O cérebro é um sistema de processamento de dados bastante complexo, paralelo e não linear. Ele consegue organizar suas estruturas básicas, os neurônios, de

modo que consiga processar certos tipos de informação bem mais rapidamente do que qualquer computador já construído (HAYKIN, 2005).

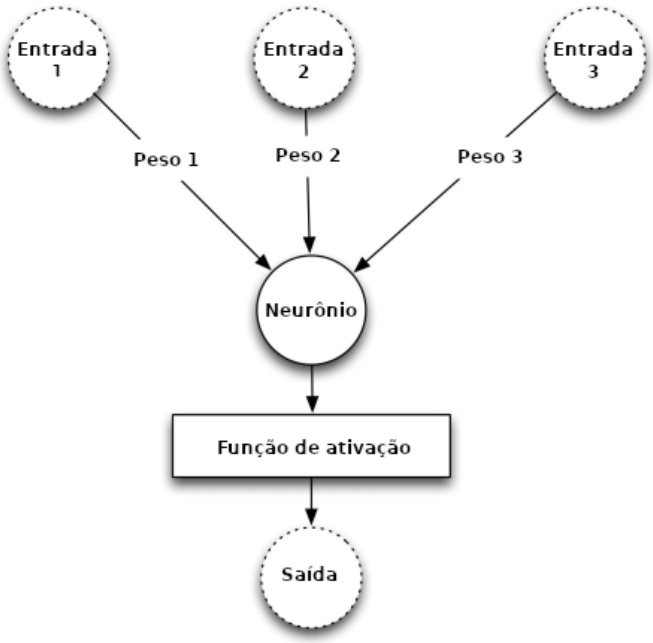
Vasilev *et al.* (2019) destacam cinco pontos importantes para o entendimento de redes neurais: o processamento de informações ocorre em elementos simples chamados neurônios; estes possuem links de conexão entre si para a troca de dados; a força de conexão entre eles pode variar, fato fundamental no processamento das informações; neurônios possuem estados internos definidos por suas conexões diretas; e apresentam funções de ativação responsáveis por determinar seu valor de saída.

Partindo de uma visão matemática, uma rede neural pode ser entendida como um grafo direcionado em que os nós correspondem aos neurônios e as arestas às ligações entre eles. Dessa forma, cada neurônio recebe como valor de entrada a soma ponderada das saídas produzidas pelos neurônios diretamente conectados a ele, e sua saída pode servir de entrada para os próximos neurônios da rede (SHALEV-SHWARTZ; BEN-DAVID, 2014).

2.5.1 Neurônio artificial

Um neurônio artificial é a unidade básica de processamento de informações dentro de uma rede neural (HAYKIN, 2005). Ele multiplica cada valor de entrada que recebe por um peso sináptico, responsável por atribuir diferentes níveis de importância a cada atributo de entrada. Posteriormente, todos esses valores são somados e aplicados a uma função de ativação, responsável por restringir a amplitude de saída do neurônio (HEATON, 2015). A Figura 6 apresenta a estrutura básica de um neurônio artificial junto de seus principais componentes.

Figura 6 – Estrutura básica e componentes principais de um neurônio artificial



Fonte: Adaptado de Heaton (2015).

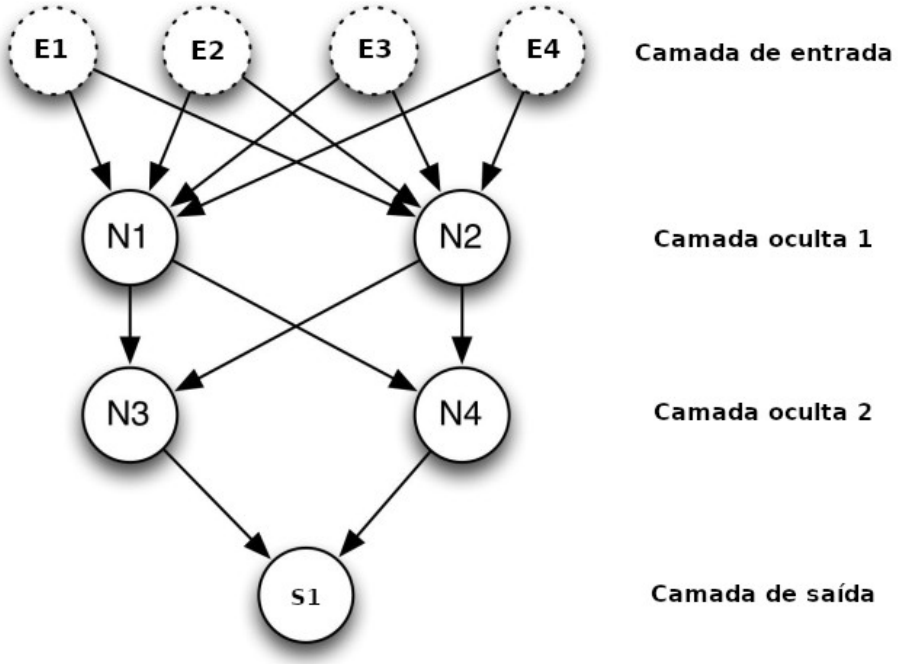
A Equação 1 corresponde a representação matemática de um neurônio artificial, em que x_i se refere aos valores de entrada do neurônio, w_i representa os pesos das ligações entre eles e b é um tipo de peso especial chamado *bias*. Por fim, o resultado é aplicado a uma etapa de transformação f, normalmente não linear, conhecida como função de ativação ou função de transferência (VASILEV *et al.*, 2019). A Seção 2.5.3 apresentará algumas das principais funções de ativação.

$$y = f(\sum_i x_i \cdot w_i + b) \tag{1}$$

2.5.2 Redes multicamadas

Uma rede neural pode apresentar uma vasta quantidade de neurônios em sua arquitetura, organizados ao longo de uma rede formada por várias camadas conectadas. A camada de entrada é a que recebe os dados, ou seja, o estado inicial do sistema. A camada de saída é a responsável por apresentar os resultados obtidos durante o processamento. As camadas extras localizadas entre a entrada e a saída do modelo são chamadas de camadas ocultas (VASILEV *et al.*, 2019). A Figura 7 apresenta um esquema geral de rede neural com duas camadas ocultas.

Figura 7 – Rede neural totalmente conectada com 2 camadas ocultas



Fonte: Adaptado de Heaton (2015).

2.5.3 Função de ativação

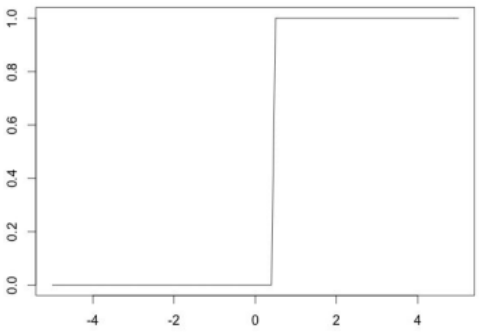
A função de ativação é um componente importante para as redes neurais artificiais, já que possibilita que uma pequena alteração nos pesos resulte apenas em uma pequena mudança na saída do modelo. É a grande responsável por determinar se a informação presente no neurônio é relevante para a rede ou se deve ser ignorada, ou seja, se o neurônio será ou não ativado. Sob uma visão matemática, consiste em uma etapa de transformação, preferencialmente não linear, dos valores de entrada para permitir a extração de informações mesmo de dados mais complexos (DATA SCIENCE ACADEMY, 2021).

Existem diversas funções de ativação disponíveis e amplamente empregadas nas mais variadas arquiteturas de redes neurais. De forma geral, mesmo uma única rede neural normalmente utilizará mais de uma função de ativação simultaneamente (TAYLOR, 2017). Alguns exemplos comuns incluem: função de etapa binária, sigmoide, tangente hiperbólica e unidade linear retificada.

A função de etapa binária, conhecida também como função baseada em limiar, é uma das funções de ativação mais simples. O neurônio é ativado apenas se o valor de entrada estiver acima de um certo valor de limiar (DATA SCIENCE ACADEMY, 2021). A Equação 2 apresenta a fórmula da função de etapa binária. Nesse caso, o limiar da função é igual a $0{,}5$, e o resultado será 1 apenas se o valor de entrada x for maior ou igual do que esse limiar, e 0 caso contrário (HEATON, 2015). A Figura 8 apresenta o comportamento gráfico da função de etapa binária.

$$f(x) = \begin{cases} 1, x \geq 0{,}5 \\ 0, x < 0{,}5 \end{cases} \tag{2}$$

Figura 8 – Representação gráfica da função de etapa binária

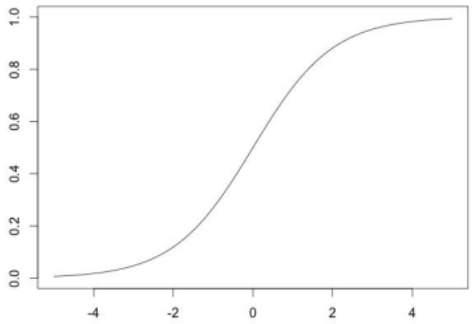


Fonte: Heaton (2015).

Já a função sigmoide, também conhecida como logística, é uma função de ativação continuamente diferenciável e não linear que limitada os valores de saída do neurônio para o intervalo entre 0 e 1. Matematicamente, pode ser interpretada como a probabilidade de ativação do respectivo neurônio (VASILEV *et al.*, 2019). A Equação 3 apresenta a fórmula da função sigmoide e a Figura 9 demonstra seu comportamento gráfico (HEATON, 2015).

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Figura 9 – Representação gráfica da função sigmoide

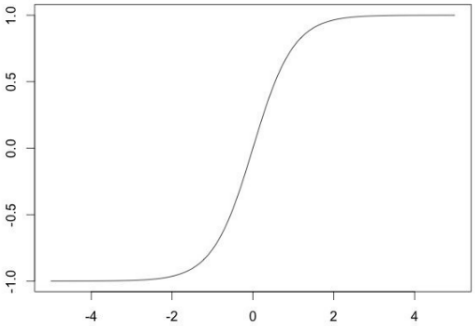


Fonte: Heaton (2015).

A função tangente hiperbólica é uma das funções de ativação mais comuns de redes neurais. Diferentemente da função sigmoide, que produz resultados apenas no intervalo de 0 a 1, a tangente hiperbólica devolve valores no intervalo que vai de -1 a 1, fazendo dela simétrica em relação ao eixo x (HEATON, 2015). Possui um comportamento contínuo e não linear, além de ser diferenciável em todos os pontos (DATA SCIENCE ACADEMY, 2021) A Equação 4 apresenta a função tangente hiperbólica (TAYLOR, 2017). A Figura 10 apresenta o comportamento gráfico da função tangente hiperbólica.

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{4}$$

Figura 10 – Representação gráfica da função de ativação tangente hiperbólica

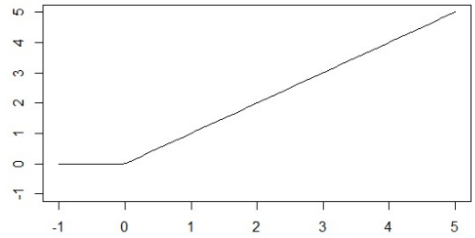


Fonte: Heaton (2015).

A função de unidade linear retificada, do inglês *Rectified Linear Unit* (ReLU), está entre as funções de ativação mais amplamente utilizadas. Dentre suas características principais, destaca-se a propriedade de ser não linear e o fato de não ativar todos os neurônios da rede simultaneamente, tornando o modelo mais esparso e facilita sua computação (DATA SCIENCE ACADEMY, 2021). A Equação 5 apresenta a fórmula matemática da função ReLU (VASILEV *et al.*, 2019). A Figura 11 demonstra o comportamento gráfico da função ReLU.

$$f(x) = \begin{cases} x, x \geq 0 \\ 0, x < 0 \end{cases} \tag{5}$$

Figura 11 – Representação gráfica da função de unidade linear retificada



Fonte: Heaton (2015).

2.5.4 Treinamento de redes neurais

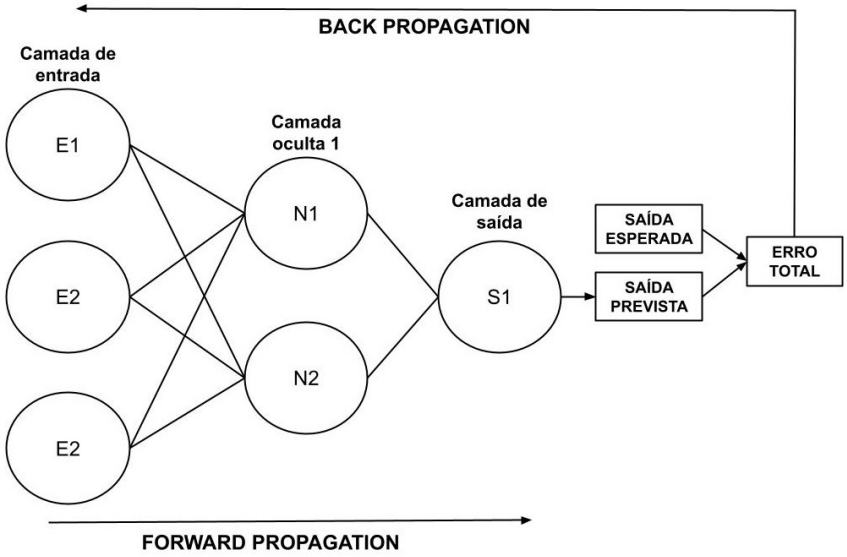
Uma rede neural pode ser considerada uma aproximação de função matemática com certo nível de erro. Nesse contexto, treinar a rede significa fazer pequenas alterações nos pesos das conexões entre os neurônios visando minimizar o erro. Em uma representação gráfica do problema em questão, o conjunto de pontos em que a função de erro é igual a zero pode ser retratada em uma hipersuperfície, e o objetivo do treinamento torna-se, a partir de um ponto, seguir uma curva na direção do valor mínimo (VASILEV *et al.*, 2019).

Em uma situação hipotética, se os recursos computacionais disponíveis para o treinamento dos modelos fossem infinitos, bastaria esgotar todas as possibilidades existentes para as configurações de pesos da rede até encontrar aquela que apresente o menor erro. No entanto, como esses recursos costumam ser bastante limitados, torna-se necessário uma solução mais inteligente para o problema, já que até mesmo redes neurais extremamente simplificadas podem apresentar uma quantidade grande de combinações de pesos (HEATON, 2015).

Um dos métodos mais utilizados para o treinamento de redes neurais é o *backpropagation*, conhecido em português como algoritmo de retropropagação, sendo um tipo especial de algoritmo de gradiente descendente (HEATON, 2015). De forma geral, o algoritmo pode ser dividido em dois passos principais: a fase *forward*, traduzida como para frente; e a fase *backward*, ou para trás (FACELI *et al.*, 2011). A Figura 12 apresenta uma representação gráfica do processo de *backpropagation*.

A fase *forward* consiste na multiplicação dos pesos de cada neurônio da primeira camada pelos valores de entrada e posterior aplicação da função de ativação. Depois, o valor resultante de cada neurônio é passado para a segunda camada, e o processo é repetido até a camada final. Por fim, a saída da última camada é comparada com o resultado esperado, e a diferença entre os valores indica o erro atual da rede (FACELI *et al.*, 2011).

Figura 12 – Esquema geral do algoritmo de *backpropagation*



Fonte: Adaptado de Taylor (2017).

O erro calculado na fase *forward* é então utilizado na segunda fase, a *backward*, por meio do cálculo de gradiente. Nesse contexto, o gradiente nada mais é do que a derivada da função de erro aplicada a cada um dos pesos da rede. Assim, o valor de gradiente de um peso da rede permite saber o quanto este deve ser modificado para diminuir o erro geral da aproximação da função (HEATON, 2015).

Dentre as abordagens existentes na literatura do algoritmo de *backpropagation*, a mais popular delas é a descida de gradiente estocástica, do inglês *Stochastic Gradient Descent* (SGD) (GOODFELLOW; BENGIO; COURVILLE, 2016). Para esse algoritmo, a aleatoriedade dos elementos do conjunto de treinamento é algo muito importante, já que a ordem dos dados poderia facilmente viciar a rede e torná-la tendenciosa (TAYLOR, 2017).

2.6 Aprendizado profundo

O aprendizado profundo, considerado uma subárea do campo de aprendizado de máquina, consiste no princípio de programar um computador de modo que ele consiga aprender sozinho, tornando-se capaz de adquirir conhecimento de forma automática e de detectar padrões escondidos em um conjunto de informações. Nesse caso, a entrada do algoritmo é formada pelos dados de treinamento, e a saída é uma especialização capaz de realizar uma tarefa (SHALEV-SHWARTZ; BEN-DAVID, 2014).

As soluções de aprendizado profundo demonstram potencial na descoberta de padrões intrínsecos mesmo em dados complexos e de alta dimensionalidade (LECUN; BENGIO; HINTON, 2015). Outro ponto importante da abordagem é a sua maior capacidade de generalizar o que foi aprendido, principalmente se comparado com as abordagens mais tradicionais de aprendizado de máquina. Isso significa que, de uma forma geral, redes neurais profundas con-

seguem lidar bem mesmo com dados nunca vistos anteriormente (GOODFELLOW; BENGIO; COURVILLE, 2016).

Essencialmente, o aprendizado profundo se baseia no uso de redes neurais profundas, que podem ser definidas como redes neurais que apresentam em sua arquitetura mais de duas camadas ocultas (HEATON, 2015). É justamente a organização da rede em várias camadas hierárquicas interconectadas que permite a compreensão e extração dos recursos dos dados em altos níveis de complexidade (VASILEV *et al.*, 2019). Quanto a função de ativação, o padrão costuma ser a função ReLU, especialmente para as camadas ocultas do modelo. Já para a camada de saída, o mais comum é o emprego de uma função de ativação linear ou sigmoide, dependendo sobretudo do objetivo do modelo em questão (HEATON, 2015).

As Redes Neurais Convolucionais, do inglês *Convolutional Neural Network* (CNN), também conhecidas como *ConvNet*, são um tipo especial de rede neural profunda. Sua aplicação está normalmente associada com tarefas que envolvem processamento de imagens digitais tais como reconhecimento e detecção de objetos. Esse tipo de rede funciona muito bem com dados organizados em uma topologia de grade, ainda mais quando existe uma forte dependência espacial entre seus componentes (AGGARWAL, 2018).

Segundo Vasilev *et al.* (2019), a dependência espacial presente em uma imagem digital está relacionada com o fato de os *pixels* que se encontram próximos estão muito mais relacionados do que os *pixels* mais distantes, e considerar essas informações em uma única unidade é muito importante. As CNNs são capazes de produzir resultados de uma mesma dimensão, mesmo lidando com entradas de dados unidimensionais, bidimensionais ou tridimensionais, trazendo uma série de vantagens no processamento de imagens digitais (VASILEV *et al.*, 2019).

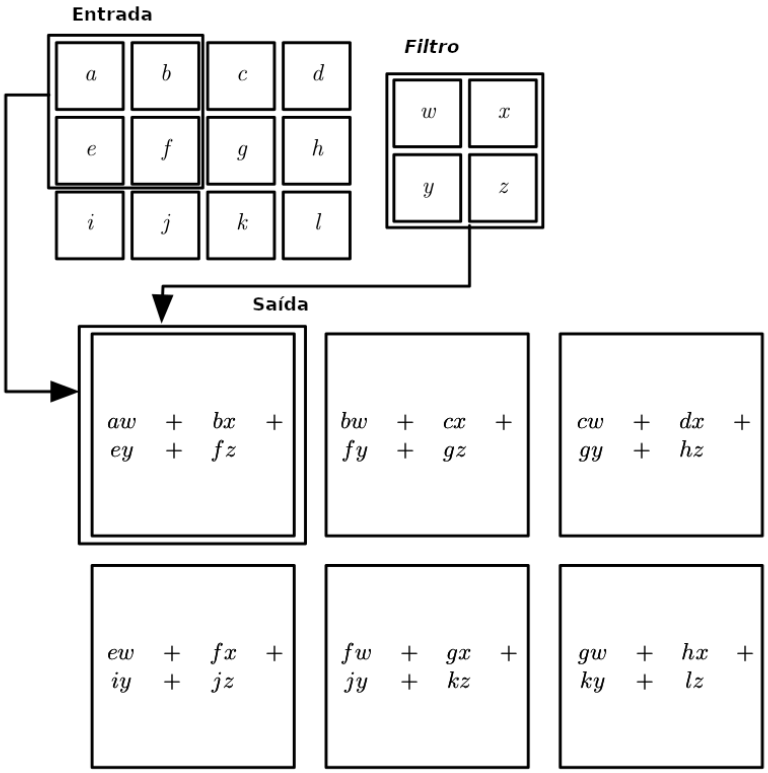
Vasilev *et al.* (2019) ainda destaca outras duas propriedades das CNNs:

- Conexão apenas entre neurônios ligados a elementos da entrada espacialmente próximos, contribuindo para o processamento das informações de regiões relacionadas e diminui o número de pesos da rede;
- Compartilhamento de pesos entre todos os neurônios de uma mesma camada, contribuindo ainda mais para a redução no número de pesos e ajudando a evitar o *overfitting*.

As CNNs usam um tipo especial de estrutura conhecida como camada de convolução, que tem por objetivo principal extrair características da imagem, tais como bordas, cores, manchas e outros elementos visuais, utilizando para isso pequenas matrizes quadradas chamadas filtros. Assim, a operação convolucional pode ser descrita como a passagem de um filtro, da esquerda para a direita, linha a linha, percorrendo cada região da entrada (HEATON, 2015). A Figura 13 apresenta graficamente a operação de convolução, em que cada região da imagem é multiplicada pelo filtro convolucional.

Após essa operação, uma camada típica desse modelo apresenta outros dois estágios. Primeiramente, cada ativação linear gerada durante a fase anterior é submetida a uma função

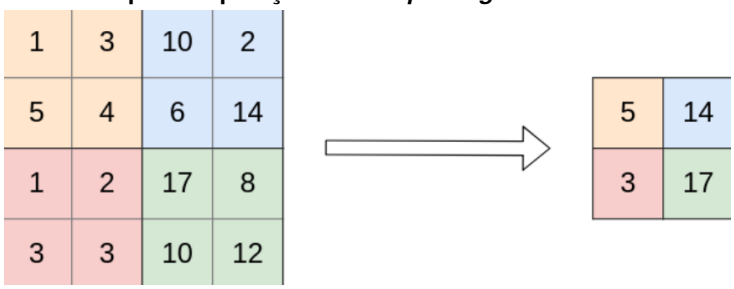
Figura 13 – Representação gráfica do processo de convolução



Fonte: Adaptado de Goodfellow, Bengio e Courville (2016).

de ativação não linear, normalmente a ReLU. Por fim, aplica-se uma operação de *pooling* ou agrupamento, responsável por transformar a saída de um local da rede em uma versão resumida das saídas próximas, contribuindo para tornar o resultado menos sensível a variações da entrada (GOODFELLOW; BENGIO; COURVILLE, 2016). O algoritmo de *pooling* mais utilizado é o *max pooling*, que preserva apenas o neurônio com o maior sinal de ativação de cada uma das regiões da entrada (VASILEV *et al.*, 2019). A Figura 14 apresenta a operação de *max pooling* com uma matriz 2×2.

Figura 14 – Exemplo de operação de *max pooling* utilizando uma matriz 2×2



Fonte: Adaptado de Vasilev *et al.* (2019).

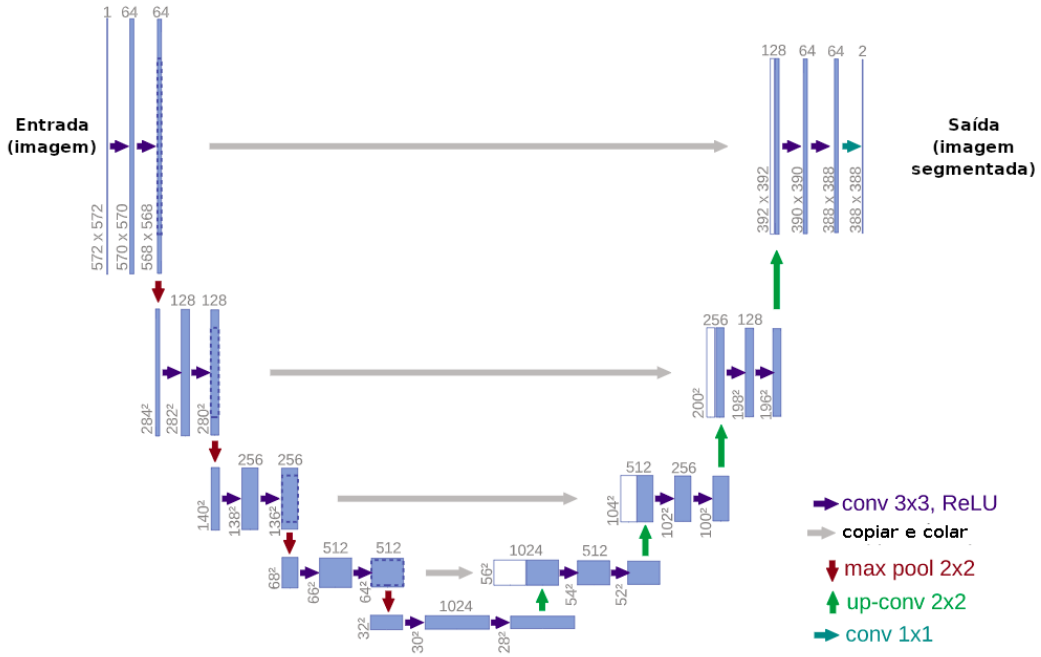
2.7 Modelos de aprendizado profundo

Diversas arquiteturas de redes neurais foram propostas na literatura ao longo dos anos, com diferentes abordagens para a resolução dos problemas mais comuns da área de aprendizado profundo aplicada à segmentação de imagens. Esta seção apresenta os modelos de redes neurais profundas relacionados com este trabalho, incluindo as redes U-Net, FPN, DeepLab, CFNet e OCRNet.

2.7.1 U-Net

A U-Net é uma rede neural convolucional profunda proposta por Ronneberger, Fischer e Brox (2015). Foi inicialmente projetada para a tarefa de segmentação semântica para aplicação em imagens biomédicas, embora seja atualmente aplicada nas mais diversas áreas de segmentação. A Figura 15 apresenta a arquitetura básica da rede U-Net.

Figura 15 – Arquitetura da rede U-Net



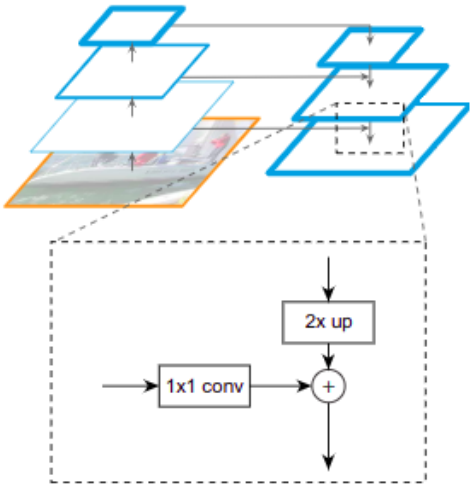
Fonte: Adaptado de Ronneberger, Fischer e Brox (2015).

A arquitetura geral da U-Net é baseada em uma estrutura em formato de "U", com duas fases principais: contração e expansão. O estágio de contração consiste em camadas de convolução, funções de ativação ReLU e operações de *pooling*. Já na segunda etapa, é aplicada uma série de operações de *upsampling* acompanhadas por convoluções e concatenações com os mapas de características obtidos na etapa anterior, o que é necessário para recuperar alguns *pixels* da borda que podem ser perdidos durante o processo. Sob outra perspectiva, a primeira etapa é responsável por extrair as características e recursos da imagem e a segunda reconstrói a imagem de forma segmentada (RONNEBERGER; FISCHER; BROX, 2015).

2.7.2 *Feature Pyramid Network* (FPN)

FPN é um modelo de aprendizado profundo proposto por Lin *et al.* (2017a). Embora tenha sido originalmente desenvolvido apenas para a detecção de objetos, também apresenta bons resultados na área de segmentação, fazendo com que o modelo também seja utilizado para essa tarefa. A Figura 16 expõe a arquitetura geral da rede FPN, com destaque para suas conexões laterais e os dois caminhos (ascendente e descendente) apresentados pelo modelo.

Figura 16 – Arquitetura geral da rede FPN



Fonte: Adaptado de Lin *et al.* (2017a).

Utiliza o formato hierárquico multiescalar natural das redes neurais convolucionais para construir pirâmides de recursos com leve custo extra. De forma geral, a arquitetura possui um caminho ascendente, um caminho descendente e diversas conexões laterais e consegue construir mapas de recursos semânticos de alto nível em todas as escalas (LIN *et al.*, 2017a). Os mapas de características gerados são então processados com a utilização de camadas de convolução 3×3, produzindo a saída de cada estágio da rede para a geração das previsões de objetos na cena. Por fim, são utilizados dois *perceptrons* multicamadas para a geração das máscaras de segmentação (MINAEE *et al.*, 2021).

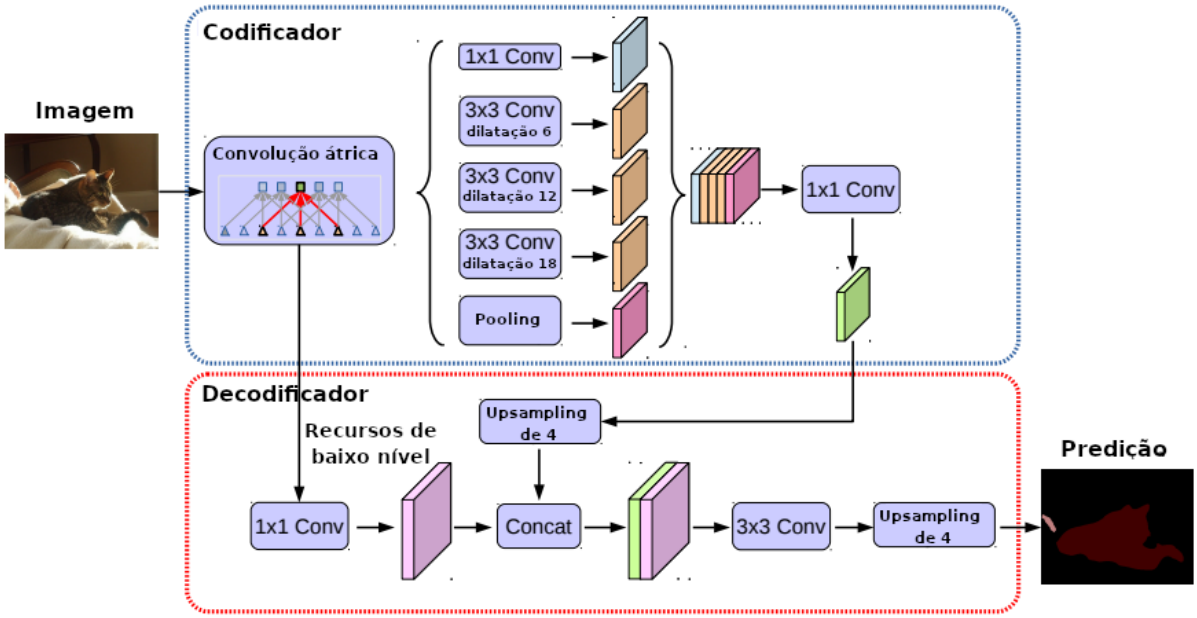
2.7.3 DeepLab

O modelo DeepLab foi proposto inicialmente por Chen *et al.* (2018a). Consiste em uma família de redes neurais profundas com a premissa fundamental de tentar solucionar dois problemas básicos das abordagens comuns da maior parte dos modelos de segmentação: primeiramente, a utilização de filtros convolucionais pequenos não consegue capturar de forma satisfatória informações contextuais das imagens; por outro lado, filtros grandes tornam os algoritmos mais lentos por conta do aumento do número de parâmetros (GHOSH *et al.*, 2019).

Para tentar solucionar os problemas, o modelo DeepLab usa três abordagens principais. Primeiramente, são utilizadas convoluções átricas (convolução dilatada), um tipo especial de filtro convolucional com fator de dilatação que permite expandir a área de visão sem aumentar o número de parâmetros. Em segundo lugar, o DeepLab usa uma estrutura conhecida como *Atrous Spatial Pyramid Pooling* (ASPP) para possibilitar a segmentação robusta em várias escalas de contexto. Por fim, são utilizadas combinações de métodos de CNNs profundas com modelos probabilísticos, possibilitando diversas melhorias na localização dos limites de objetos na imagem (CHEN *et al.*, 2018a).

A versão DeepLabv3+, proposta em Chen *et al.* (2018b), implementa ainda mais algumas melhorias em comparação ao modelo original. A primeira delas é a adição de um módulo decodificador simples e eficaz, refinando os resultados do modelo especialmente ao longo das bordas dos objetos. Por segundo, houve a aplicação de convolução separável em profundidade para os módulos ASPP e decodificadores, resultando em desempenho melhor e mais rápido. A Figura 17 apresenta a arquitetura do modelo DeepLabv3+ (CHEN *et al.*, 2018b).

Figura 17 – Arquitetura geral da rede DeepLabv3+



Fonte: Adaptado de Chen *et al.* (2018b).

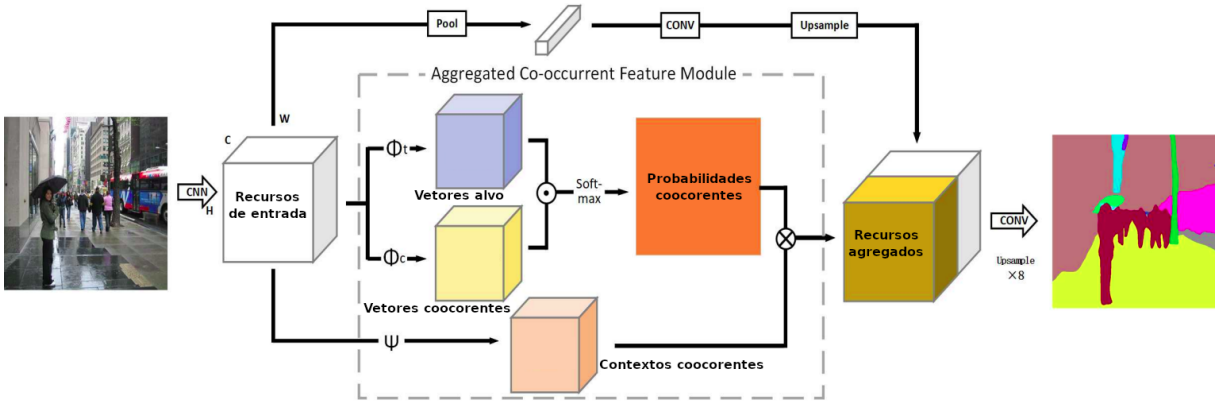
2.7.4 CFNet

CFNet é um modelo de rede neural profunda desenvolvido por Zhang *et al.* (2019). É fortemente baseado no conceito de coocorrência, ou seja, na inter-relação existente entre os objetos em uma cena. O uso da coocorrência consegue melhorar a robustez de um sistema e contribuir para a eliminação de possíveis ambiguidades nos rótulos de objetos, fator normalmente relacionado com a presença de ruídos e variações na posição de objetos ou de ilumina-

ção. o CFNet propõe a modelagem da coocorrência de traços na forma de uma distribuição de probabilidade, chamada *Aggregated Co-ocorrent Feature Module* ou ACF (ZHANG *et al.*, 2019).

A Figura 18 apresenta a visão geral da arquitetura da rede CFNet. A partir de uma imagem de entrada, ocorre a extração dos mapas de recursos por CNNs pré-treinadas. Na sequência, os mapas de recursos são transformados em representações de vetores alvo e vetores de coocorrência. O modelo calcula as probabilidades de coocorrência a partir das semelhanças entre os pares vetoriais. Em seguida, o módulo ACF integra o resultado anterior com o contexto de coocorrência, enquanto outra ramificação utiliza de operações de convolução para obter as características globais. Por fim, a etapa final consiste nas previsões ao nível de *pixel* dos rótulos que dão origem a imagem segmentada (ZHANG *et al.*, 2019).

Figura 18 – Arquitetura geral da rede CFNet



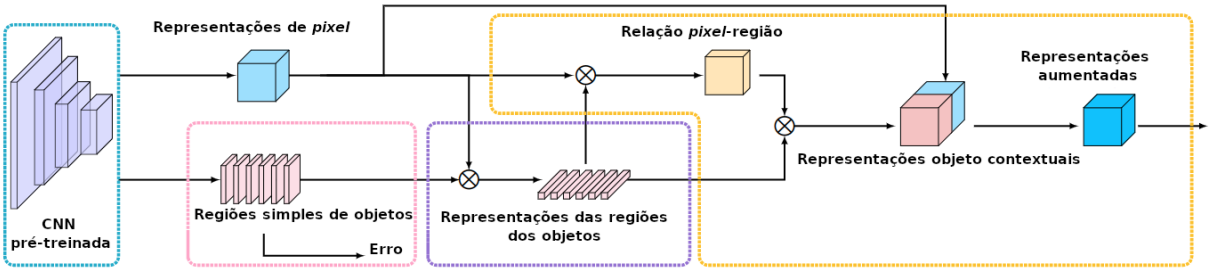
Fonte: Adaptado de Zhang *et al.* (2019).

2.7.5 OCRNet

O modelo OCR, do inglês *Object-Contextual Representation*, foi proposto por Yuan *et al.* (2020). É baseado no princípio de representação objeto-contextual, de forma que a classificação de um *pixel* se dá a partir da exploração de sua representação da classe de objeto correspondente. A abordagem geral da rede se concentra no cálculo da representação da área de um objeto a partir da agregação dos *pixels* de sua região, para posterior cálculo da relação entre cada *pixel* e cada região do objeto (YUAN *et al.*, 2020).

A Figura 19 apresenta a arquitetura geral do modelo OCRNet. Primeiramente, ocorre a divisão dos *pixels* da imagem de entrada em um conjunto simples de regiões por CNNs pré-treinadas (regiões azul e rosa da Figura 19). Depois, é feita a estimativa da representação de cada região a partir da agregação dos *pixels* dos objetos correspondentes (área roxa da Figura 19). Por fim, ocorre o aumento da representação de cada *pixel* com a representação contextual do objeto, a partir de uma agregação ponderada das representações de um objeto com os pesos calculados a partir das relações espaciais dos *pixels* e objetos na cena (região amarela na Figura 19) (YUAN *et al.*, 2020).

Figura 19 – Arquitetura geral da rede OCRNet



Fonte: Adaptado de Yuan *et al.* (2020).

2.8 Métricas para a avaliação de desempenho

Para que um sistema de segmentação possa produzir uma contribuição significativa para um determinado campo, seu desempenho precisa ser rigorosamente avaliado a partir de um conjunto de métricas definidas e padronizadas, que permitam comparações justas com resultados já estabelecidos na literatura (GARCIA-GARCIA *et al.*, 2017). Essa avaliação pode ocorrer sob diferentes aspectos, conforme o propósito ou contexto do sistema em questão. De forma geral, os aspectos mais utilizados incluem a acurácia e a eficiência (MINAEE *et al.*, 2021).

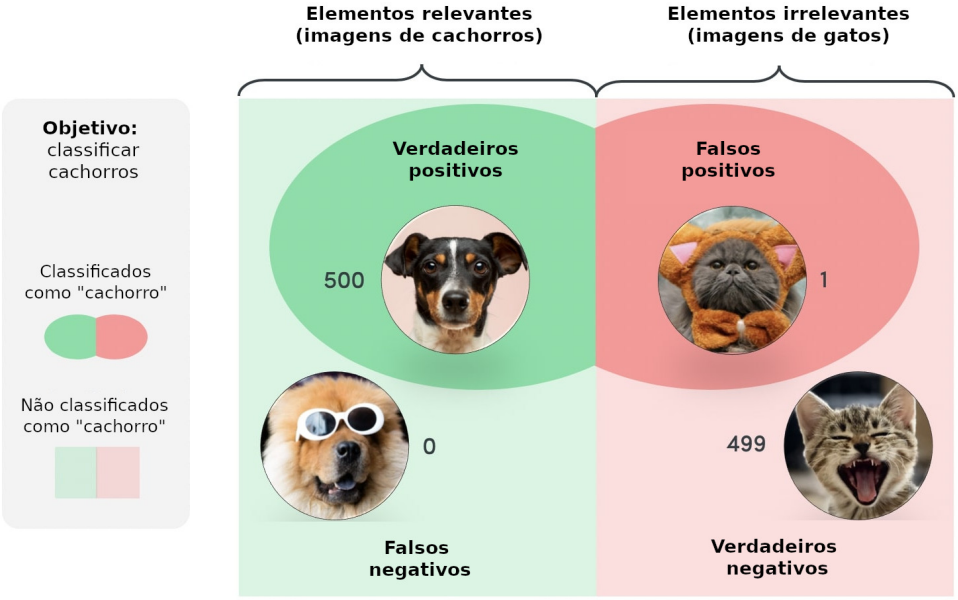
Outra forma de promover uma comparação justa dos resultados, além de também aumentar a quantidade de dados representativos disponíveis para o treinamento de modelos de redes neurais, é a criação de conjuntos de dados públicos para atender as principais aplicações de aprendizado de máquina. Para a tarefa de segmentação, um dos *datasets* mais populares e utilizados na literatura é o *Pascal Visual Object Classes* (Pascal VOC), um conjunto de dados constituído de 21 classes e mais de 3.000 imagens.

2.8.1 Acurácia

Dentre as medidas de avaliação de desempenho, a acurácia é certamente uma das mais populares e utilizadas na literatura. Nesse sentido, diversas métricas para avaliação da acurácia dos modelos de segmentação foram propostas pelos pesquisadores da área ao longo dos anos. Dentre essas métricas, destacam-se: precisão, cobertura, *F-score* e Interseção sobre União (IoU) (MINAEE *et al.*, 2021).

Para o cálculo das métricas para a classe positiva, são utilizados os conjuntos de *pixels* classificados corretamente para a classe positiva (verdadeiros positivos ou VP), os atribuídos incorretamente para a classe positiva (falsos positivos ou FP), os que pertencem à classe positiva mas não foram atribuídos a ela (falsos negativos ou FN) e os que foram classificados corretamente como não pertencentes a classe positiva (verdadeiros negativos ou VN). A Figura 20 apresenta um exemplo para um problema de classificação de cachorros em um conjunto de dados que contêm gatos e cachorros.

Figura 20 – Possíveis conjuntos para um problema de classificação de cachorros



Fonte: Adaptado de Huellmann (2022).

A precisão ou *precision* consiste no número de *pixels* positivos classificados corretamente em relação a todos os *pixels* classificados como positivos (FACELI *et al.*, 2011). Sob outra perspectiva, a precisão descreve o quão bom um modelo é na tarefa de identificar corretamente os objetos em uma imagem (ATIENZA, 2020). A Equação 6 apresenta a fórmula para cálculo da precisão (FACELI *et al.*, 2011).

$$\textit{Precisão} = \frac{VP}{VP + FP} \tag{6}$$

Já a cobertura, do inglês *recall*, consiste no número de *pixels* positivos classificados corretamente em relação a todos os *pixels* que realmente pertencem à classe positiva (FACELI *et al.*, 2011). De forma geral, a cobertura é uma boa métrica para indicar o quão bom o modelo é em identificar todos os casos relevantes. A Equação 7 apresenta a fórmula matemática para cálculo da cobertura (ATIENZA, 2020).

$$\textit{Cobertura} = \frac{VP}{VP + FN} \tag{7}$$

Uma precisão igual a $1{,}0$ para uma classe P indica que todos os *pixels* classificados como pertencentes a classe P realmente pertencem a ela. Uma cobertura igual a $1{,}0$ indica que todos os *pixels* pertencentes a classe P foram atribuídos para ela. Por conta dessa natureza complementar, essas duas métricas acabam sendo combinadas em uma única medida conhecida como *F-score* ou medida-F (FACELI *et al.*, 2011). Sob um ponto de vista matemático, a *F-score* pode ser definida como a média harmônica entre a precisão e a cobertura. A Equação 8 apresenta a fórmula matemática para o cálculo da *F-score*, em que precisão e cobertura possuem o mesmo peso (F_1) (MINAEE *et al.*, 2021).

$$\textit{F-score} = \frac{2 \cdot precisão \cdot cobertura}{precisão + cobertura} \tag{8}$$

A medida de Interseção sobre União, do inglês *Intersection over Union* (IoU), também conhecida como *Jaccard Index*, pode ser definida como o número de *pixels* positivos classificados corretamente divididos pela soma de todos os *pixels* que realmente pertencem à classe positiva com os *pixels* atribuídos incorretamente para a classe positiva. Sob outro ponto de vista, o IoU mede a quantidade de *pixels* comuns entre a segmentação correta e o resultado do modelo em relação ao total de *pixels* compartilhados entre ambos os casos. É uma das métricas de avaliação de desempenho que mais se destaca na literatura, devido à sua capacidade de representatividade aliada a simplicidade. A Equação 9 expõe a fórmula matemática para cálculo da IoU (GARCIA-GARCIA *et al.*, 2017).

$$IoU = \frac{VP}{VP + FN + FP} \tag{9}$$

2.8.2 Eficiência

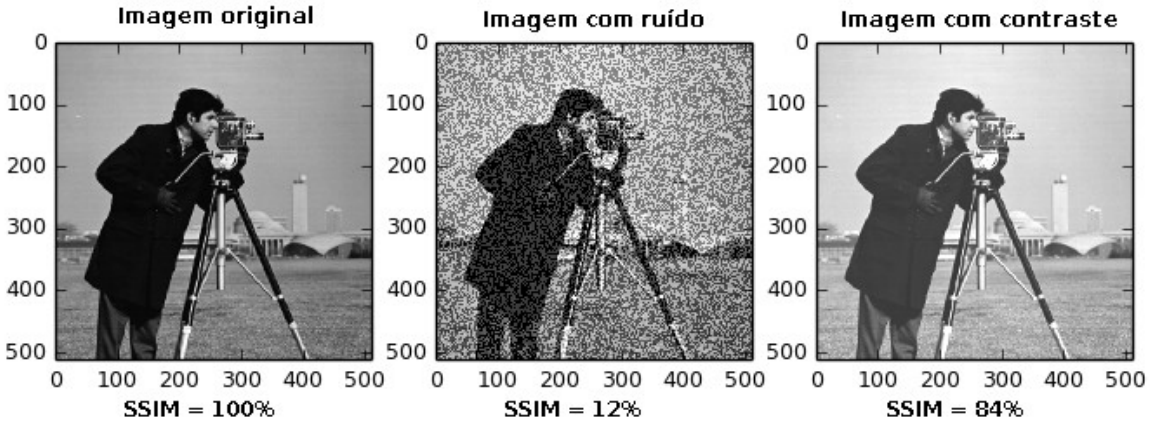
A velocidade ou tempo de execução é uma métrica bastante útil, visto que a maioria das aplicações deve atender a requisitos rígidos de consumo de tempo. Em alguns casos, também pode ser útil informar o tempo gasto com o treinamento, questão intimamente relacionada com a reprodutibilidade do experimento. Com o tempo, é comum a descrição completa do hardware e do ambiente utilizado, visto que estes podem influenciar fortemente no desempenho do sistema (GARCIA-GARCIA *et al.*, 2017).

O uso de memória é outro fator importante para a avaliação de modelos de segmentação. Dependendo da aplicação, a quantidade de armazenamento exigida pode ser um aspecto limitante, como no caso de sistemas embarcados ou dispositivos móveis. Também pode fazer sentido considerar a quantidade de memória necessária para o treinamento do modelo, fator que está mais relacionado com a reprodutibilidade e viabilidade de implementação do sistema (GARCIA-GARCIA *et al.*, 2017).

2.8.3 Similaridade

Existem diversas abordagens na literatura utilizadas com a finalidade de quantificar o grau de semelhança (ou diferença) entre imagens digitais. Um dos métodos mais utilizados para esse fim é o *Structural Similarity Index* (SSIM), um índice usado para estimar a similaridade estrutural entre duas imagens. O índice varia no intervalo de 1 até -1, em que 1 indica que duas imagens são idênticas quanto a sua estrutura, 0 aponta diferença total e -1 sugere que as duas imagens apresentam estruturas invertidas (DOSSELMANN; YANG, 2011; WANG *et al.*, 2004). A Figura 21 apresenta um exemplo de aplicação do SSIM.

Figura 21 – Exemplo de aplicação do SSIM



Fonte: Adaptado de Scikit-Image (2021).

A Equação 10 apresenta a fórmula matemática para cálculo do SSIM, em que: x e y correspondem as duas imagens que devem ser comparadas; $C1$ e $C2$ são constantes usadas para estabilizar a conta no caso de valores muito próximos de zero; μ_x e μ_y representam as médias de x e y, respectivamente; σ_x e σ_y a variância de x e y, respectivamente; e por fim σ_{xy} indica a covariância de x e y (WANG *et al.*, 2004).

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C2)}{(\mu_x^2 + \mu_y^2 + C1)(\sigma_x^2\sigma_y^2 + C2)} \tag{10}$$

2.9 Dados sintéticos

O aprendizado profundo conseguiu promover uma revolução de alto nível nas mais diversas áreas, com destaque para a visão computacional e o processamento de linguagem natural. Grande parte desse sucesso está relacionado com os avanços no poder computacional (principalmente com a evolução das unidades de processamento gráfico), bem como as melhorias nos algoritmos, o desenvolvimento de novas abordagens para o treinamento dos modelos e o *big data* (MELO *et al.*, 2022).

Apesar de todo o seu potencial, os modelos de aprendizado profundo apresentam uma séria limitação: dependem de uma quantidade abundante de dados anotados prontamente disponíveis para a otimização de seus parâmetros, até mesmo para compreender conceitos relativamente simples (MELO *et al.*, 2022). Segundo Nikolenko (2021), as fases de obtenção e anotação dos dados para o treinamento de modelos de aprendizado profundo podem representar cerca de 80% do tempo gasto com qualquer projeto real da área, que dependa de informações que não estão prontamente disponíveis em *datasets* públicos.

Uma possível solução para o problema é a geração de conjuntos de dados sintéticos. A técnica consiste na criação de exemplos artificiais que imitam ou se aproximam dos dados reais. Exemplos sintéticos costumam ser consideravelmente mais simples e rápidos de serem

obtidos do que os dados reais, além de serem inesgotáveis e pré-anotados. O uso de dados sintéticos também pode ajudar a prevenir dilemas éticos, como no caso de dados sigilosos, ou ser útil em situações em que a coleta pode ser impraticável ou envolver questões de segurança (MELO *et al.*, 2022).

Patki, Wedge e Veeramachaneni (2016) comprovaram por meio de uma série de experimentos que o treinamento de modelos de aprendizado profundo utilizando apenas informações sintéticas possui o potencial de atingir resultados tão bons quanto os obtidos usando dados reais, dependendo apenas da complexidade do problema e do método empregado para a geração dessas informações. Outra abordagem recorrente é a de utilizar técnicas de geração de dados sintéticos como uma forma de aumentar a quantidade de exemplares disponíveis para o treinamento, o que pode proporcionar uma melhoria significativa dos resultados obtidos (FAWAZ *et al.*, 2018).

Existem diversas abordagens de geração de dados sintéticos para problemas de visão computacional, desde as mais simples que utilizam apenas técnicas de processamento de imagem, até as mais complexas que usam redes generativas adversárias (GANs) (NIKOLENKO, 2021). Uma abordagem bastante funcional é a de geração de imagens sintéticas pela fusão de diferentes fontes de dados, a partir da sobreposição de objetos na cena. Também conhecida como composição de imagem, a técnica baseia-se na combinação de objetos, como pessoas ou animais, sob diferentes configurações e fundos, garantindo uma maior variabilidade das amostras (MELO *et al.*, 2022). A Figura 22 exemplifica esse procedimento, em que a imagem de um cachorro foi combinada com uma paisagem de fundo para dar origem a uma nova imagem sintética.

Figura 22 – Exemplo de processo de composição de imagens

Fonte: Patrawala (2020).

2.10 Trabalhos correlatos

O problema de segmentar uma imagem digital já foi bastante abordado a partir da utilização de técnicas mais tradicionais de visão computacional e de aprendizado de máquina. Embora esses métodos sejam populares e capazes de alcançar resultados interessantes, sua habilidade de resolver desafios mais complexos é limitada. A evolução das técnicas de aprendizado profundo foi um marco em muitos problemas de visão computacional, incluindo as tarefas de segmentação de imagens digitais (GARCIA-GARCIA *et al.*, 2017).

Ngugi, Abelwahab e Abo-Zahhad (2020) conduziram diversos testes visando segmentar folhas de tomate. Um *dataset* próprio foi construído para os experimentos, consistindo de 1.408 imagens de folhas de tomate capturadas em condições de campo desafiadoras, tais como: grandes variações de iluminação, incluindo imagens com alta insolação e outras tiradas no período noturno com auxílio de iluminação artificial; fundo complicado, com presença de múltiplas folhas, sombras, frutos e solo; e múltiplas câmeras com diferentes resoluções. O *dataset* foi dividido em 70% das imagens para treinamento, 20% para validação e 10% para teste.

Os modelos utilizados por Ngugi, Abelwahab e Abo-Zahhad (2020) incluem as redes SegNet e U-Net. Devido à variabilidade de topologias que esses dois modelos podem ser implementados, vários testes foram realizados visando escolher as mais apropriadas para a tarefa em questão. A rede U-Net com 5 estágios de codificador e 64 filtros por camada de convolução se sobressaiu perante as demais arquiteturas, alcançando 95,82% de acurácia média. Dentre as arquiteturas SegNet, a rede com 4 estágios codificadores e 128 filtros por camada de convolução apresentou o melhor desempenho, com uma acurácia média de 94,63%.

Gonçalves *et al.* (2021) fizeram uma série de experimentos visando a segmentação semântica de folhas saudáveis e doentes. As fotos foram obtidas de três diferentes *datasets*: o primeiro[1] contem 406 imagens de café arábica apresentando lesões de bicho mineiro; o segundo[2] possui 208 exemplares de soja com manchas de ferrugem; e o terceiro[3] conta com 152 imagens de folhas com mancha de trigo. Para o treinamento dos modelos, os *datasets* passaram por processo de anotação manual das regiões nas classes fundo, saudável e doente.

Gonçalves *et al.* (2021) utilizaram um total de seis arquiteturas de redes neurais convolucionais: U-Net, SegNet, *Pyramid Scene Parsing Network* (PSPNet), FPN e DeepLabv3+ (com duas variações). O conjunto de dados foi dividido aleatoriamente, com 80% das imagens destinadas ao treinamento dos modelos e 20% aos testes. Independentemente da arquitetura, os melhores resultados para precisão, cobertura e IoU foram obtidos na segmentação das folhas de fundo, seguidos das saudáveis e das doentes. As arquiteturas FPN, U-Net e DeepLabv3+ obtiveram os melhores desempenhos dentre os modelos utilizados.

[1] https://osf.io/ygq82/
[2] https://osf.io/4hbr6/
[3] https://osf.io/52cjn

Esgario *et al.* (2021) propuseram em seu artigo uma abordagem composta de dois estágios testados separadamente e em diferentes redes neurais convolucionais: o primeiro estágio busca a segmentação semântica com cálculo da severidade da infecção foliar enquanto a segunda etapa foca na classificação de lesões. Por fim, um aplicativo móvel foi construído para permitir a utilização dos modelos treinados em campo por fazendeiros. O *dataset* BRACOL[4] foi utilizado no experimento, um conjunto de dados com 1.747 imagens de folhas de café arábica saudáveis ou apresentando um ou mais sintomas de bicho mineiro, cercospora, phoma e ferrugem, tiradas em condições de laboratório.

Para o estágio de segmentação semântica, Esgario *et al.* (2021) utilizaram duas arquiteturas de redes neurais, U-Net e PSPNet, treinadas em 500 imagens manualmente anotadas do *dataset*. Os resultados obtidos indicaram um desempenho superior da rede U-Net (94,85% de IoU) frente a PSPNet (93,69% de IoU). Além disso, a rede U-Net também foi consideravelmente mais eficiente do que a PSPNet, tanto considerando o tempo de treinamento dos modelos quanto o de inferência de resultados.

2.11 Considerações finais

Este capítulo apresentou diversos conceitos importantes relacionados ao trabalho, tais como os aspectos da cultura do café e seus principais distúrbios, fundamentos de imagens digitais, aprendizado de máquina, redes neurais e aprendizado profundo, além de princípios de segmentação de imagens e o estado da arte do tema. No próximo capítulo, serão abordados os materiais e os métodos que serão utilizados no decorrer do trabalho.

[4] https://data.mendeley.com/datasets/yy2k5y8mxg/1

3 MATERIAIS E MÉTODOS

Após os estudos feitos sobre o café e suas características, além da análise dos princípios de segmentação de imagens digitais e principais técnicas de aprendizado profundo aplicadas para este fim, é importante definir os procedimentos que serão adotados para alcançar os objetivos propostos. Dessa forma, este capítulo se destina a apresentação dos materiais e métodos utilizados ao longo do desenvolvimento deste trabalho.

3.1 Materiais

Esta seção busca descrever os materiais utilizados no decorrer do projeto, como as configurações de hardware e ambiente de desenvolvimento escolhido, a linguagem de programação, as bibliotecas e os conjuntos de dados.

3.1.1 Ambiente de desenvolvimento

Ao longo de todo o desenvolvimento do projeto, optou-se pela utilização do *Python*[1], uma linguagem de programação de alto nível, interpretada, dinâmica, multiplataforma e de código aberto, amplamente utilizada para a construção de soluções de inteligência artificial, aprendizado de máquina e ciência de dados. *Python* também oferece uma série de bibliotecas prontas muito úteis para manipulação de dados e redes neurais (PYTHON BRASIL, 2021).

O ambiente principal adotado é o *Google Colaboratory*[2], também chamado *Colab*, um ambiente *online* e interativo que permite escrever código *Python* do navegador. O *Colab* também possibilita o uso de *Graphics Processing Unit* (GPUs) e *Tensor Processing Unit* (TPUs), capazes de acelerar bastante o processo de treinamento de redes neurais profundas. Para ter acesso a GPUs mais poderosas, o plano *pro* foi utilizado, o que representou um custo de R$ 58,00 por mês ao longo de 4 meses de experimentos. As especificações técnicas do ambiente incluem:

- *Central Processing Unit* (CPU) de dois núcleos *Xeon* 2,2 GHz;
- GPU Tesla P100-PCIE 16 GB;
- 13,6 GB de memória RAM e 166,83 GB de armazenamento em disco;
- Python 3.7.12.

[1] https://www.python.org/
[2] https://colab.research.google.com/

3.1.2 Bibliotecas e ferramentas

Diversas bibliotecas foram utilizadas para facilitar e agilizar os testes no decorrer deste trabalho. *Tensorflow*[3] é um ecossistema flexível, abrangente e *open source* de bibliotecas e ferramentas para a área de aprendizado de máquina. Permite a criação e o treinamento de redes neurais profundas de forma facilitada e com poucas linhas de código, além de possibilitar a utilização de GPUs e TPUs capazes de reduzir consideravelmente o tempo de treinamento (TENSORFLOW, 2021).

Matplotlib[4] é uma biblioteca *Python* utilizada para a criação e visualização de gráficos e imagens, principalmente devido a sua simplicidade e facilidade de uso (MATPLOTLIB, 2021). Undouble[5] é uma biblioteca *Python* construída para auxiliar na detecção e remoção de imagens idênticas ou parecidas, utilizando para isso uma série de etapas de pré-processamento (escala de cinza, normalização e dimensionamento) e cálculo de *hash* (TASKESEN, 2020).

Albumentations[6] é uma biblioteca *Python* eficiente e flexível que oferece uma extensa variedade de operações de transformação de imagem, utilizadas em tarefas de visão computacional. As aplicações mais comuns da biblioteca incluem o aumento de dados para tarefas de aprendizado profundo e competições de aprendizado de máquina, além de diversos projetos de código aberto (ALBUMENTATIONS, 2021). *Computer Vision Annotation Tool* (CVAT)[7] é uma ferramenta *online*, gratuita e interativa para anotação de imagens e vídeos, possuindo suporte a uma ampla variedade de formatos para as anotações (CVAT, 2021).

Segmentation Models[8], desenvolvida por Yakubovskiy (2019), é uma biblioteca *Python* baseada em *Tensorflow* que conta com diversas implementações de redes neurais profundas para a segmentação de imagens, incluindo os modelos U-Net e FPN, citados na Seção 2.7. *TensorFlow Advanced Segmentation Models*[9], desenvolvida por Kezmann (2020), é uma biblioteca fortemente inspirada na *Segmentation Models* que conta com a implementação de outros modelos, incluindo DeepLabv3+, CFNet e OCRNet, também citados na Seção 2.7.

3.1.3 Bases de dados

Para a construção do conjunto de dados sintéticos, foi necessário buscar por bases de dados públicas com imagens que pudessem ser utilizadas como novo plano de fundo para as imagens sintéticas geradas. Dessa forma, diversos bancos de imagens disponíveis na Internet

[3] https://tensorflow.org/
[4] https://matplotlib.org/
[5] https://github.com/erdogant/undouble
[6] https://albumentations.ai/
[7] https://cvat.org/
[8] https://github.com/qubvel/segmentation_models/
[9] https://github.com/JanMarcelKezmann/TensorFlow-Advanced-Segmentation-Models/

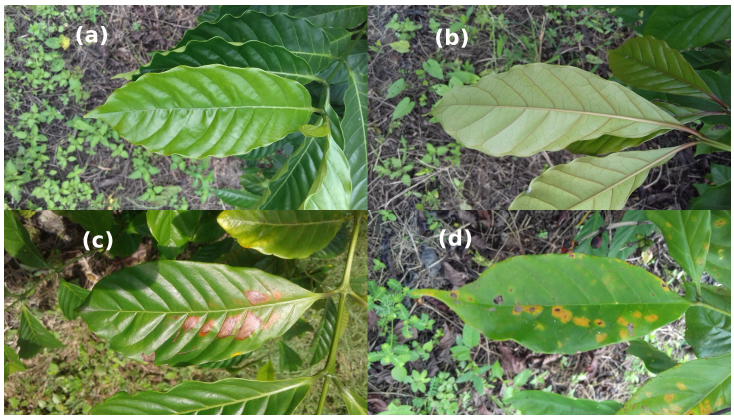
foram utilizados, tais como *Pexels*[10], *Unsplash*[11], *IStock*[12], *Pixabay*[13], *FreeImages*[14], *Burst*[15] e o Repositório *Digipathos*[16]. Desses conjuntos, foram selecionadas 200 imagens que apresentam características relacionadas com imagens de campo, tais como terra, grama, árvores, plantações e outras. A Figura 23 apresenta alguns exemplos das imagens utilizadas. Vale ressaltar que todos os repositórios permitem o uso das imagens para fins acadêmicos.

Figura 23 – Exemplos usados como plano de fundo das imagens sintéticas

Fonte: Adaptado de Barbedo *et al.* (2018).

Quanto às imagens de folhas de café, o primeiro conjunto de dados utilizado foi produzido por Parraga-Alava *et al.* (2019) e possui 1.560 imagens de café da espécie conilon com resoluções variando de 2048×1152 a 4128×2322 *pixels*. O conjunto contém imagens de folhas saudáveis e outras apresentando um ou mais sintomas de ferrugem e ácaro vermelho. As fotos foram capturadas no campo em condições reais, incluindo variações de luminosidade (manhã e tarde, dias ensolarados e nublados) e diferentes planos de fundo (outras plantas, terra, ervas daninhas). A Figura 24 apresenta alguns exemplos do conjunto de dados.

Figura 24 – Exemplos do primeiro conjunto de dados. (a) e (b) folha saudável; (c) ácaro vermelho; (d) ferrugem

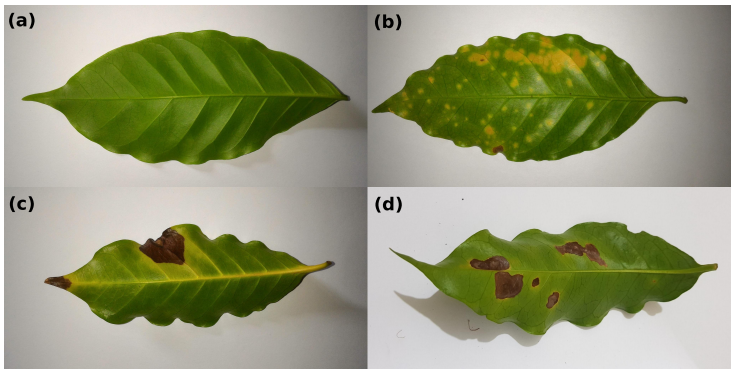


Fonte: Adaptado de Parraga-Alava *et al.* (2019).

[10] https://www.pexels.com/
[11] https://unsplash.com/
[12] https://www.istockphoto.com/br
[13] https://pixabay.com/pt/
[14] https://www.freeimages.com/
[15] https://burst.shopify.com/
[16] https://www.digipathos-rep.cnptia.embrapa.br/

A segunda base de dados usada foi desenvolvida por Esgario, Krohling e Ventura (2020) e conta com 1.747 imagens de café da espécie arábica com resolução de 2048×1024 *pixels*, sendo que 500 dessas imagens já incluem máscara de segmentação de folha e fundo. O *dataset* contém folhas saudáveis e outras apresentando um ou mais sintomas de bicho mineiro, cercospora, phoma e ferrugem. As imagens foram capturadas com uma variedade de câmeras e em diferentes épocas do ano para garantir maior variabilidade dos dados, e todas foram capturadas em laboratório. A Figura 25 apresenta alguns exemplos do conjunto de dados.

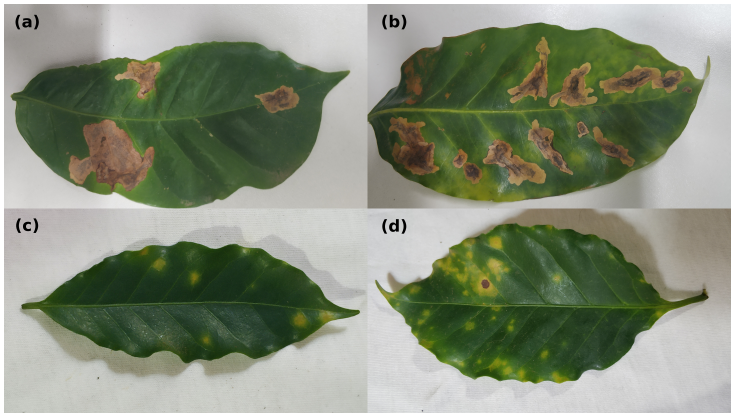
Figura 25 – Exemplos do segundo conjunto de dados. (a) folha saudável; (b) ferrugem; (c) phoma; (d) bicho mineiro



Fonte: Adaptado de Esgario, Krohling e Ventura (2020).

O terceiro *dataset* utilizado foi criado por Silva, Carneiro e Faulin (2020) e conta com um total de 539 imagens de café da espécie arábica com resolução de 4000×2250 *pixels*. Nenhuma segmentação prévia foi fornecida com o conjunto de dados. As folhas apresentam sintomas de bicho mineiro e de ferrugem em diversos estágios de contaminação. Todas as imagens foram coletadas com uma câmera de *smartphone* e em ambiente de laboratório. A Figura 26 apresenta alguns exemplos do conjunto de dados.

Figura 26 – Exemplos do terceiro *dataset*. (a) e (b) bicho mineiro; (c) e (d) ferrugem

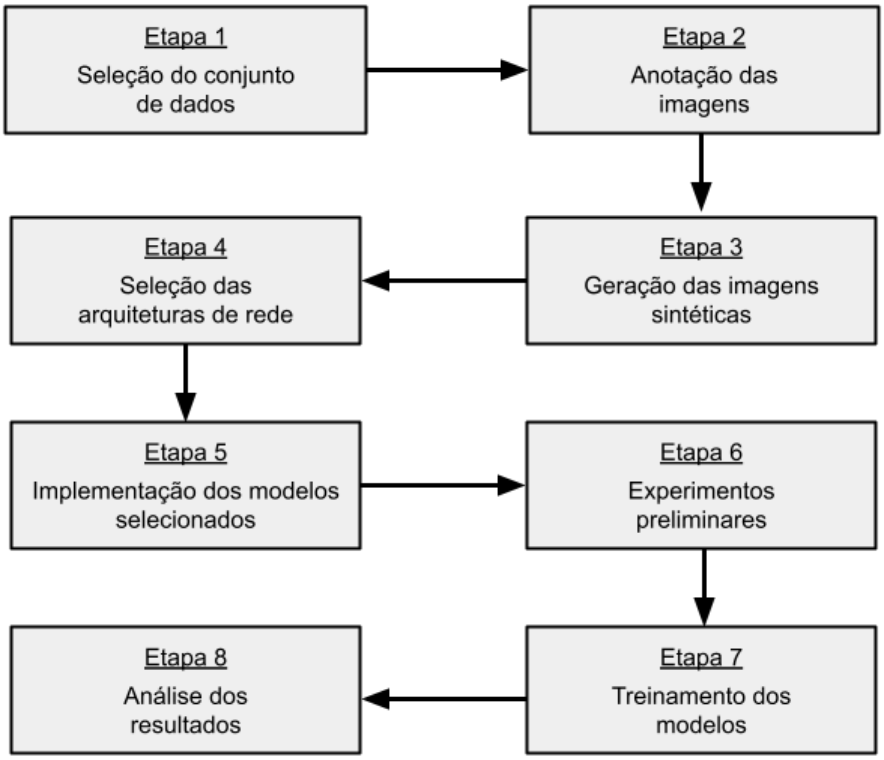


Fonte: Adaptado de Silva, Carneiro e Faulin (2020).

3.2 Métodos

Com o intuito de melhorar a organização e o fluxo geral de trabalho do projeto, optou-se pela divisão do conjunto de tarefas em oito estágios principais de desenvolvimento, que vão desde a seleção dos conjuntos de dados utilizados até a análise dos resultados obtidos pelos modelos de segmentação. O fluxograma da Figura 27 sintetiza a ideia, apresentando o conjunto de etapas executadas com a finalidade de alcançar os objetivos propostos.

Figura 27 – Fluxograma da organização geral do projeto



Fonte: Autoria própria (2022).

3.2.1 Primeira etapa: seleção do conjunto de dados

A primeira etapa consistiu na busca por *datasets* públicos de folhas de café. Como especificado na Seção 3.1.3, foram escolhidos três conjuntos de dados principais da cultura de interesse, totalizando 3.846 imagens. As bases de dados selecionadas incluem tanto imagens tiradas em laboratório quanto imagens de campo das duas principais espécies de café citadas nas Seções 2.1.1 e 2.1.2. A Tabela 1 sintetiza as informações gerais do conjunto de dados resultante. Além disso, também foram selecionadas 200 imagens de diversos conjuntos de dados disponíveis na Internet para serem utilizadas como plano de fundo para as imagens sintéticas.

Tabela 1 – Síntese de informações dos conjuntos de dados usados no trabalho

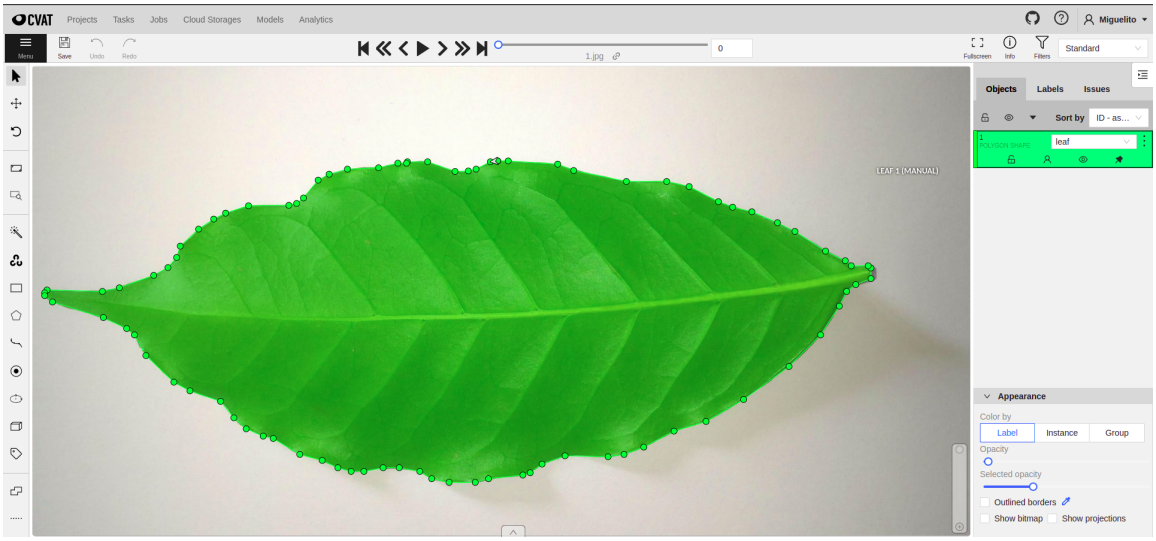
Conjuntos de dados	Nº de imagens de laboratório	Nº de imagens de campo	Total de imagens
Parraga-Alava *et al.* (2019)	0	1.560	1.560
Esgario, Krohling e Ventura (2020)	1.747	0	1.747
Silva, Carneiro e Faulin (2020)	539	0	539
Total de imagens	2.286	1.560	3.846

Fonte: Autoria própria (2022).

3.2.2 Segunda etapa: anotação das imagens

A segunda etapa consistiu no processo de rotulação dos dados. Cada imagem foi segmentada em regiões conforme as classes disponíveis: fundo ou folha. Dentre as diversas ferramentas gratuitas disponíveis para o processo de rotulação, optou-se pela utilização do CVAT, principalmente por conta de sua simplicidade e conjunto de ferramentas descritas na Seção 3.1.2. Essencialmente, a rotulação no CVAT consistiu na especificação dos pontos que compõe as extremidades dos elementos da classe *leaf* (folha), enquanto os *pixels* restantes de cada imagem pertencem à classe fundo. A Figura 28 apresenta a interface do CVAT, com destaque para a segmentação de uma folha.

Figura 28 – Interface do CVAT com destaque para processo de segmentação



Fonte: Autoria própria (2022).

Antes da segmentação manual, o conjunto de dados desenvolvido por Parraga-Alava *et al.* (2019) precisou passar por uma etapa de remoção de imagens semelhantes, uma vez que elas podem atrapalhar no treinamento dos modelos de aprendizado profundo. Para isso, foi utilizada a biblioteca Undouble, que examinou e comparou cada imagem do *dataset* em busca de exemplares com alto grau de similaridade. As imagens semelhantes foram analisadas visualmente utilizando a biblioteca Matplotlib, resultando na remoção de 106 imagens. Do conjunto restante, foram segmentadas manualmente 638 imagens utilizando o CVAT.

Para facilitar o processo de anotação das imagens de laboratório dos *datasets* desenvolvidos por Esgario, Krohling e Ventura (2020) e Silva, Carneiro e Faulin (2020), optou-se pela geração das máscaras faltantes a partir do treinamento do modelo de segmentação U-Net, por tratar-se de uma rede consolidada e eficiente. Para isso, foram utilizados como dados de treinamento as 500 imagens do *dataset* Esgario, Krohling e Ventura (2020) que já estavam segmentadas. O modelo resultante permitiu a geração das 1.786 máscaras restantes, sendo necessária apenas uma revisão para detectar possíveis inconsistências e promover melhorias nas anotações geradas. A Tabela 2 sintetiza as informações de anotação das imagens.

Tabela 2 – Síntese de informações de anotação das imagens

Conjuntos de dados	Anotações prévias	Anotações manuais	Anotações geradas	Total de anotações
Parraga-Alava *et al.* (2019)	0	638	0	638
Esgario, Krohling e Ventura (2020)	500	0	1.247	1.747
Silva, Carneiro e Faulin (2020)	0	0	539	539
Total de anotações	500	638	1.786	2.924

Fonte: Autoria própria (2022).

3.2.3 Terceira etapa: geração das imagens sintéticas

A terceira etapa englobou o procedimento de geração do conjunto de dados sintéticos utilizando as 2.286 imagens de laboratório dos *datasets* de Esgario, Krohling e Ventura (2020) e Silva, Carneiro e Faulin (2020). O objetivo desta etapa foi, a partir de imagens tiradas em laboratório com condições controladas de fundo e iluminação, obter exemplares que apresentam características mais próximas daquelas encontradas em imagens reais de campo, para aumentar a quantidade de dados relevantes disponíveis para o treinamento dos modelos neurais.

Essa etapa pode ser dividida em 3 momentos principais. Primeiramente, foi realizado o ajuste das 2.286 imagens de folhas de café, processo que consistiu na padronização das resoluções para que correspondessem com o tamanho alvo do *dataset* sintético (2.048×2.048) e aplicação de transparência ao fundo a partir das máscaras de segmentação utilizando o canal alfa das imagens. A Figura 29 faz uma comparação do antes (esquerda) e depois (direita) de uma imagem do *dataset*.

Figura 29 – Folha original (esquerda) e folha pré-processada (direita)

Fonte: Autoria própria (2022).

Depois, foi realizado o ajuste das 200 imagens do fundo, processo que consistiu na padronização da resolução das imagens para o intervalo de 2.048×2.048 até 3.072×3.072. O objetivo foi dar flexibilidade para que o fundo sintético apresente pequenas mudanças mesmo nos casos em que uma mesma imagem seja utilizada mais de uma vez, mas sem comprometer sua representatividade. Também foi feita a checagem do *dataset* visando eliminar imagens muito semelhantes, utilizando para isso as biblioteca Undouble e Matplotlib, citadas na Seção 3.1.2.

Por fim, foi feita a geração das imagens sintéticas a partir dos dois conjuntos pré-processados de folhas e fundos. Uma ou mais folhas eram escolhidas aleatoriamente e passaram por uma série de transformações espaciais, incluindo rotação, espelhamento (horizontal e vertical), movimentação pelos eixos (até 50% do tamanho original da imagem), variação de escala (de 60% a 90% do tamanho original da imagem) e de inclinação (até 30º). Do outro conjunto, uma imagem de fundo era escolhida ao acaso e passava por transformações espaciais que incluíam recorte aleatório para a resolução alvo (2.048×2.048) e espelhamento (horizontal e vertical). Por fim, as imagens selecionadas eram combinadas formando uma nova imagem sintética. A Figura 30 apresenta um exemplo de imagem sintética gerada.

Figura 30 – Exemplo de imagem gerada (esquerda) e sua máscara de segmentação (direita)

Fonte: Autoria própria (2022).

3.2.4 Quarta etapa: seleção das arquiteturas

A tarefa seguinte consistiu na seleção das arquiteturas de redes neurais treinadas para a tarefa de segmentação. Foram escolhidas cinco redes neurais profundas: U-Net, por ser uma arquitetura tradicional na tarefa de segmentação associado ao fato de apresentar um bom desempenho, como comprovado por Esgario, Krohling e Ventura (2020); FPN, a partir dos bons resultados obtidos por Gonçalves *et al.* (2021); e os modelos DeepLabv3+ (CHEN *et al.*, 2018b), CFNet (ZHANG *et al.*, 2019) e OCRNet (YUAN *et al.*, 2020), três redes que se destacaram consideravelmente por conta dos bons resultados que obtiveram no *dataset* Pascal VOC, mencionado na Seção 2.8.

3.2.5 Quinta etapa: implementação dos modelos selecionados

A implementação dos modelos foi feita utilizando-se as bibliotecas *Segmentation Models* e *TensorFlow Advanced Segmentation Models*, citadas na Seção 3.1.2, por possuírem interfaces de alto nível para a configuração e o treinamento dos modelos, o que facilita consideravelmente o processo e minimiza a possibilidade de erros de implementação. As bibliotecas também contam com diversas CNNs pré-treinadas no conjunto de dados ImageNet[17], um grande banco de imagens destinado a tarefas de reconhecimento de objetos. O uso de CNNs pré-treinadas está relacionado com o conceito de transferência de aprendizado, uma técnica de aprendizado profundo em que uma rede neural treinada em um determinado conjunto de dados (como o ImageNet) é utilizada para melhorar a generalização do conhecimento em outro conjunto de dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

As bibliotecas utilizadas também facilitam a etapa de configuração de alguns aspectos importantes para o treinamento de modelos de aprendizado profundo, tais como a função de perda, a função de custo e o otimizador. Nesse sentido, a função de perda utilizada foi a *focal loss*, proposta por Lin *et al.* (2017b), uma generalização da entropia cruzada binária que busca reduzir o efeito de desbalanceamento dos dados de treinamento. Todos os experimentos também utilizaram a função de ativação sigmoide para a camada de saída dos modelos e o otimizador Adam, proposto por Kingma e Ba (2014).

Além da etapa de geração dos dados sintéticos, realizada com a finalidade de expandir a quantidade de imagens relevantes disponíveis para o treinamento, também foram implementadas outras técnicas de aumento de dados. Essas técnicas consistem na aplicação de diferentes transformações espaciais e de *pixel* às imagens durante o processo de treinamento dos modelos, com o objetivo de promover pequenas variações nos dados de modo a aumentar a variabilidade dos exemplos disponíveis para o treinamento. A biblioteca *Albumentations* foi utilizada para a implementação das transformações nas imagens durante o treinamento dos modelos. A Seção 3.2.6 apresenta mais detalhes das diferentes transformações aplicadas às imagens.

3.2.6 Sexta etapa: experimentos preliminares

Diversas variáveis podem afetar o desempenho de modelos de aprendizado profundo, incluindo fatores externos (como o *dataset* utilizado) e internos à rede (como a resolução das imagens, a taxa de aprendizado, o tamanho do lote de treinamento e a CNN pré-treinada utilizada). Testar todas as combinações possíveis de variáveis para todos os modelos não é uma tarefa viável. A solução normalmente adotada para esse problema é a implementação de experimentos preliminares, que consistem em um conjunto de treinamentos destinados ao teste das configurações mais comuns, de modo a encontrar aquelas que apresentam o melhor desempenho em relação aos objetivos propostos neste trabalho.

[17] https://image-net.org/

Dessa forma, diversos testes foram conduzidos visando definir os hiperparâmetros utilizados no treinamento dos modelos. A rede escolhida para essa etapa foi a U-Net, por tratar-se de um modelo bastante consolidado e que apresenta um treinamento mais rápido do que as demais alternativas mencionadas na Seção 3.2.4. A rede ResNet-50, proposta por He *et al.* (2015), foi utilizada como CNN pré-treinada, por conta de sua eficiência e bom desempenho se comparada a outras opções disponíveis nas bibliotecas utilizadas. Além disso, os hiperparâmetros gerais utilizados incluem taxa de aprendizado de 10^{-4}, tamanho do lote igual a 16, resolução de 512×512 e 50 épocas de treinamento, podendo variar de acordo com o objetivo do experimento em questão.

Todas as imagens de folhas de laboratório pré-processadas e todas as imagens de fundo pré-processadas (2.286 e 200, respectivamente) foram usadas na geração do *dataset* sintético do conjunto de treinamento. Já os conjuntos de validação e de teste foram formados por 64 imagens reais de campo cada um, anotadas do *dataset* de Parraga-Alava *et al.* (2019). Para avaliação do desempenho dos modelos, foram aplicadas as métricas precisão, cobertura, *F-score* e IoU, citadas na Seção 2.8.1. Os experimentos preliminares propostos são:

- **Experimento Preliminar 1:** variações das configurações utilizadas na geração do *dataset* sintético.
- Experimento Preliminar 1.1: geração de 571 imagens para o conjunto de treinamento, correspondendo a quarta parte da quantidade de folhas disponíveis para esse fim. Processamento idêntico ao descrito na Seção 3.2.3;
- Experimento Preliminar 1.2: semelhante ao Experimento 1.1, mas além do processamento descrito na Seção 3.2.3, houve a adição de etapas de compressão (50 a 70% da qualidade original), desfoque (Gaussiano, da mediana e de movimento, com o tamanho do filtro variando entre 5 e 7) e distorção (por *grid* e óptica) das imagens de fundo, visando uma aproximação maior do *dataset* gerado em comparação com as imagens reais.
- **Experimento Preliminar 2:** variações das técnicas de aumento de dados aplicadas durante o treinamento utilizando a biblioteca *Albumentations*.
- Experimento Preliminar 2.1: aplicação de transformações espaciais às imagens, incluindo espelhamento (horizontal e vertical), movimentação pelos eixos (de até 10% do tamanho da imagem), variação de perspectiva (5 a 10%) e escala (de 80 a 120% do tamanho original);
- Experimento Preliminar 2.2: além dos processamentos citados no Experimento 2.1, foram acrescentadas transformações ao nível de *pixel*, incluindo adição de ruído gaussiano, equalização, e variações de até 20% dos valores originais de brilho, contraste e saturação das imagens;

- Experimento Preliminar 2.3: além dos processamentos descritos no Experimento 2.2, foram acrescentadas algumas transformações ao nível de *pixel*, incluindo a geração de sombras (um ou dois polígonos de tamanho e posição aleatória que escurecem regiões da imagem) e variações aleatórias na relação de áreas claras e escuras das imagens (até 10% do valor original).

- **Experimento Preliminar 3:** variações da taxa de aprendizado utilizada no treinamento entre os valores 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2} e 10^{-1} (Experimentos Preliminares 3.1, 3.2, 3.3, 3.4 e 3.5, respectivamente).

- **Experimento Preliminar 4:** variações da resolução das imagens entre os valores 224×224, 256×256, 448×448 e 512×512 (Experimentos Preliminares 4.1, 4.2, 4.3 e 4.4, respectivamente).

- **Experimento Preliminar 5:** variações nas configurações de treinamento da CNN pré-treinada.

- Experimento Preliminar 5.1: os pesos da CNN pré-treinada foram congelados e usados apenas como extratores de características, alterado-se apenas os pesos das novas camadas da rede durante o treinamento;

- Experimento Preliminar 5.2: o treinamento foi feito com todos os parâmetros treináveis descongelados. Nesse caso, tanto os pesos do modelo pré-treinado quanto os das novas camadas da rede serão modificados.

- **Experimento Preliminar 6:** variações no tamanho do *dataset* sintético utilizado.

- Experimento Preliminar 6.1: geração de 571 imagens sintéticas para o treinamento, correspondendo a quarta parte da quantidade de folhas disponíveis para esse fim;

- Experimento Preliminar 6.2: geração de 1.143 imagens sintéticas para o treinamento, correspondendo a metade da quantidade de folhas disponíveis para esse fim;

- Experimento Preliminar 6.3: geração de 2.286 imagens sintéticas para o conjunto de treinamento, correspondendo a quantidade total de folhas disponíveis para esse fim.

3.2.7 Sétima etapa: treinamento dos modelos

A sétima etapa correspondeu ao treinamento efetivo dos modelos previamente selecionados mencionados na Seção 3.2.4 com os hiperparâmetros que obtiveram os melhores resultados nos testes preliminares citados na Seção 3.2.6. O conjunto de dados utilizado em todos os experimentos consiste de 1.143 imagens sintéticas destinadas 100% para o treinamento e as 638 imagens anotadas do *dataset* Parraga-Alava *et al.* (2019) divididas em 90% para

treinamento (510 imagens), 5% para validação (64 imagens) e 5% para testes (64 imagens), totalizando 1.781 imagens. Para avaliação do desempenho dos modelos, foram utilizadas as métricas precisão, cobertura, *F-score* e IoU, citadas na Seção 2.8.1.

O treinamento de todos os modelos foi definido para ser executado por até 500 épocas, com parada antecipada caso o erro de validação não diminua por mais de 70 épocas. O objetivo foi fornecer épocas suficientes de treinamento para modelos que precisam de mais tempo para convergir sem prolongar desnecessariamente os treinamentos mais rápidos. Além disso, foi configurada uma condição de redução automática da taxa de aprendizado em um fator de 0,1 caso o erro de validação não diminua por 30 épocas seguidas, o que pode beneficiar o treinamento dos modelos em caso de estagnação do aprendizado. A abordagem adotada para o tamanho do lote de treinamento consistiu na utilização do maior valor suportado pelo *Colab* (em potência de dois), o que depende de fatores como o modelo de segmentação ou a CNN pré-treinada utilizada. Cada modelo de segmentação foi treinado quatro vezes, cada treinamento utilizando uma CNN pré-treinada diferente como extratora de características, sendo:

- **ResNet-50:** proposto por He *et al.* (2015), é um modelo muito utilizado na tarefa de classificação ou como extrator de características para tarefas de segmentação, principalmente por conta de seu bom desempenho, baixo consumo de memória e boa velocidade de treinamento. Possui acurácia máxima de 77,1% no conjunto de dados ImageNet (PAPERS WITH CODE, 2022);

- **ResNet-152:** também proposto por He *et al.* (2015), é uma variação do modelo ResNet-50, mas apresentando mais parâmetros. Isso torna seu treinamento mais lento e pesado, mas potencializa seus resultados. Apresenta acurácia máxima de 78,57% no conjunto de dados ImageNet (PAPERS WITH CODE, 2022);

- **DenseNet-121:** proposto por Huang *et al.* (2016), se baseia em uma estrutura substancialmente mais profunda que as abordagens mais tradicionais, além de contar com conexões mais curtas entre as camadas próximas da entrada e da saída. Alcançou uma acurácia máxima de 74,98% no conjunto de dados ImageNet (PAPERS WITH CODE, 2022);

- **EfficientNet-B3:** proposto por Tan e Le (2019), baseia-se em um método de dimensionamento uniforme de suas dimensões (profundidade, largura e resolução), resultando em mais eficiência mesmo utilizando uma quantidade menor de parâmetros. Possui acurácia máxima de 81,1% no conjunto de dados ImageNet (PAPERS WITH CODE, 2022).

4 RESULTADOS E DISCUSSÃO

Este capítulo se destina a apresentação dos resultados obtidos nos experimentos preliminares propostos para a definição dos hiperparâmetros, bem como a análise dos resultados de treinamento dos modelos de segmentação descritos na Seção 3.2.4. Para a avaliação do desempenho dos modelos, optou-se pela atribuição de uma importância maior para as medidas IoU e *F-score*, mencionadas na Seção 2.8, por se tratarem de métricas mais representativas para a análise e comparação dos resultados.

4.1 Resultados dos experimentos preliminares

Esta seção busca apresentar e analisar os resultados obtidos do conjunto de experimentos preliminares que foram propostos com o objetivo de testar e definir os melhores hiperparâmetros, para posterior aplicação nos treinamentos dos demais modelos de aprendizado profundo propostos no escopo deste trabalho.

4.1.1 Experimento Preliminar 1

O primeiro conjunto de experimentos preliminares teve como objetivo principal comparar duas diferentes abordagens de pré-processamento das imagens utilizadas como plano de fundo no momento de geração do conjunto de dados sintéticos. A Tabela 3 sintetiza os resultados obtidos utilizando as métricas apresentadas na Seção 2.8. Como esperado, o Experimento 1.2 (fundo processado) apresentou o melhor desempenho, sendo 1,36 pontos percentuais (pp) de IoU e 0,77 pp de *F-score* acima do desempenho do Experimento 1.1.

Tabela 3 – Resultados obtidos no conjunto 1 de experimentos preliminares

Experimento	**Imagens de fundo**	**IoU (%)**	**Precisão (%)**	**Cobertura (%)**	***F-score* (%)**
1.1	Original	70,72	91,51	76,21	82,39
1.2	Processado	72,08	92,37	76,67	83,16

Fonte: Autoria própria (2022).

As transformações aplicadas ao plano de fundo das imagens sintéticas no Experimento 1.2 (borramento, distorção e compressão) foram capazes de contribuir para o aprendizado do modelo e melhorar seu desempenho, pois estão relacionadas com características frequentemente encontradas no plano de fundo das imagens reais de campo, o que certamente contribui para que o modelo se acostume com elas.

4.1.2 Experimento Preliminar 2

O segundo conjunto de experimentos preliminares teve como objetivo principal verificar a importância da aplicação de etapas de aumento de dados no conjunto de imagens sintéticas durante o treinamento dos modelos. De uma forma geral, a técnica tende a melhorar a representatividade geral do conjunto de dados e minimizar o risco de *overfitting*. A Tabela 4 sintetiza os resultados obtidos nos experimentos.

Tabela 4 – Resultados obtidos no conjunto 2 de experimentos preliminares

Experimento	Aumento de dados	IoU (%)	Precisão (%)	Cobertura (%)	*F-score* (%)
2.1	Básico	78,74	90,40	86,27	87,67
2.2	Completo	80,13	89,73	88,43	88,59
2.3	Avançado	80,98	89,79	89,39	89,18

Fonte: Autoria própria (2022).

A aplicação dos procedimentos de aumento de dados no *pipeline* de treinamento melhorou consideravelmente os resultados obtidos, principalmente em comparação com o desempenho apresentado pelos Experimentos Preliminares 1.1 e 1.2. O Experimento preliminar 2.3, que reuniu a maior quantidade de transformações espaciais e de *pixel*, foi o que obteve os melhores resultados, conquistando 0,85 pp de IoU e 0,59 pp de *F-score* a mais do que o Experimento 2.2, além de 2,24 pp de IoU e 1,51 pp de *F-score* a mais que o Experimento 2.1.

4.1.3 Experimento Preliminar 3

O terceiro conjunto de experimentos preliminares teve como objetivo principal verificar o comportamento do modelo conforme a variação da taxa de aprendizado definida para o treinamento. A Tabela 5 apresenta as métricas de cada experimento no conjunto de testes. Valores intermediários para a taxa de aprendizado foram os que obtiveram os melhores resultados, com destaque para Experimento 3.2 (10^{-4}), que apresentou o maior valor tanto de IoU (80,98%) quanto de *F-score* (89,18%), cerca de 1,38 e 0,98 pp a mais do que as mesmas métricas do segundo experimento com o melhor desempenho (Experimento 3.3).

Tabela 5 – Resultados obtidos no conjunto 3 de experimentos preliminares

Experimento	Taxa de aprendizado	IoU (%)	Precisão (%)	Cobertura (%)	*F-score* (%)
3.1	10^{-5}	77,74	87,27	88,16	87,09
3.2	10^{-4}	80,98	89,79	89,39	89,18
3.3	10^{-3}	79,60	92,21	85,53	88,20
3.4	10^{-2}	76,82	88,91	85,08	86,13
3.5	10^{-1}	69,78	75,52	89,58	80,91

Fonte: Autoria própria (2022).

De uma forma geral, valores muito baixos para esse hiperparâmetro (como 10^{-5}) tornam o aprendizado do modelo lento e aumentam as chances de que ele fique preso em ótimos locais. Por outro lado, valores muito altos (como no caso de 10^{-1}) provocam alterações grandes e desgovernadas nos pesos da rede, tornando-a instável. Dessa forma, valores mais intermediários (como 10^{-4} ou 10^{-3}) costumam ser as melhores opções para casos como esse.

4.1.4 Experimento Preliminar 4

O quarto conjunto de experimentos preliminares teve como objetivo principal verificar o desempenho do modelo sob diferentes resoluções de imagem. As resoluções escolhidos foram as mais comuns na literatura, incluindo os valores normalmente utilizados para o treinamento das CNNs no conjunto ImageNet (224×224 e 256×256) e resoluções um pouco maiores, como 448×448 e 512×512. A Tabela 6 apresenta o desempenho de cada experimento. Os melhores resultados foram obtidos utilizando-se resoluções maiores para as imagens, com destaque para o Experimento Preliminar 4.4, que obteve ganhos de 5,47 pp de IoU e 3,60 pp de *F-score*, em comparação com o experimento com a menor resolução testada (Experimento 4.1).

Tabela 6 – Resultados obtidos no conjunto 4 de experimentos preliminares

Experimento	Resolução (*pixels*)	IoU (%)	Precisão (%)	Cobertura (%)	*F-score* (%)
4.1	224×224	75,51	86,32	85,89	85,58
4.2	256×256	77,03	86,63	87,49	86,62
4.3	448×448	80,78	89,31	89,54	88,97
4.4	512×512	80,98	89,79	89,39	89,18

Fonte: Autoria própria (2022).

O melhor desempenho obtido pelo Experimento Preliminar 4.4 pode ser justificado pelo fato de resoluções maiores preservarem melhor os detalhes da imagem original, que acabam sendo prejudicados ou até perdidos quando são usadas resoluções baixas. Por outro lado, o treinamento de modelos neurais com resoluções altas demais possui um custo computacional alto e obriga a redução do tamanho do lote de treinamento, fatores que impossibilitaram o uso de resoluções maiores do que 512×512 neste trabalho.

4.1.5 Experimento Preliminar 5

O quinto conjunto de experimentos preliminares teve como objetivo principal verificar o comportamento do modelo de aprendizado profundo quanto ao congelamento ou não dos pesos da CNN pré-treinada utilizada no treinamento para a transferência de aprendizado. A partir da observação dos resultados apresentados na Tabela 7 é possível perceber que o modelo treinado no Experimento Preliminar 5.2, com o ajuste de todos os parâmetros da rede (pesos

descongelados), conseguiu aprender melhor a segmentar as folhas de café, apresentando um ganho de 1,06 pp de IoU e 0,57 pp de *F-score*.

Tabela 7 – Resultados obtidos no conjunto 5 de experimentos preliminares

Experimento	Pesos da rede	IoU (%)	Precisão (%)	Cobertura (%)	*F-score* (%)
5.1	Congelados	80,98	89,79	89,39	89,18
5.2	Descongelados	82,04	92,46	88,09	89,75

Fonte: Autoria própria (2022).

Treinar um modelo de aprendizado profundo utilizando uma CNN pré-treinada com todos os pesos treináveis descongelados costuma permitir um melhor ajuste da rede ao problema em questão, resultando em ganhos de desempenho, como observado nos resultados deste experimento. No entanto, alguns problemas podem surgir em decorrência disso: mais dados costumam ser necessários para permitir o melhor ajuste dos parâmetros e o risco de que o modelo sofra de *overfitting* é potencialmente mais elevado.

4.1.6 Experimento Preliminar 6

O sexto conjunto de experimentos preliminares teve como objetivo principal verificar a influência da quantidade de imagens sintéticas geradas nos resultados obtidos pelo modelo de aprendizado profundo. A Tabela 8 apresenta os resultados obtidos, com destaque para o Experimento Preliminar 6.2, que conseguiu ganhos de 0,47 pp de IoU e 0,36 pp de *F-score* em comparação ao Experimento 6.1, que obteve o segundo melhor desempenho.

Tabela 8 – Resultados obtidos no conjunto 6 de experimentos preliminares

Experimento	Nº de imagens (treinamento)	IoU (%)	Precisão (%)	Cobertura (%)	*F-score* (%)
6.1	571	82,04	92,46	88,09	89,75
6.2	1.143	82,51	88,89	92,12	90,11
6.3	2.286	81,96	88,78	91,79	89,74

Fonte: Autoria própria (2022).

Intuitivamente, espera-se que o aumento na quantidade de dados de treinamento melhore o desempenho do modelo, ainda mais quando todos os pesos treináveis da rede são descongelados. Entretanto, não foi o que aconteceu com o Experimento Preliminar 6.3, treinado com a maior quantidade de dados sintéticos. Possivelmente, o aumento na quantidade de folhas geradas nesse experimento não conseguiu acrescentar exemplos realmente novos e relevantes para o treinamento, visto que a quantidade base de imagens de folhas e fundos é limitada e igual a utilizada nos outros experimentos.

4.2 Resultados dos treinamentos

Esta seção visa apresentar e analisar os resultados dos modelos de segmentação U-Net, FPN, DeepLabv3+, CFNet e OCRNet, utilizando as variações de CNNs pré-treinadas ResNet-50, ResNet-152, DenseNet-121 e EfficientNet-B3. Os padrões adotados para todos os treinamentos, definidos após a avaliação dos resultados dos experimentos preliminares, incluem: fundo processado para as imagens sintéticas; aumento de dados avançado; taxa de aprendizado de 10^{-4}; resolução de 512×512; descongelamento dos pesos da CNN pré-treinada; e conjunto de dados sintéticos com 1.143 imagens. A Tabela 9 apresenta o desempenho obtido pelos modelos de segmentação, com destaque em negrito para os três melhores resultados de cada métrica apresentada.

Tabela 9 – Desempenho dos modelos de segmentação no conjunto de testes

Modelo	Extrator de características	IoU (%)	Precisão (%)	Cobertura (%)	*F-score* (%)
U-Net	ResNet-50	85,99	92,59	92,44	92,14
FPN	ResNet-50	86,95	**94,30**	91,86	92,68
DeepLabv3+	ResNet-50	85,62	89,81	**95,00**	91,95
CFNet	ResNet-50	86,26	92,78	92,65	92,34
OCRNet	ResNet-50	84,17	91,58	91,61	91,10
U-Net	ResNet-152	85,90	92,69	92,29	92,13
FPN	ResNet-152	85,82	93,30	91,77	92,01
DeepLabv3+	ResNet-152	85,74	89,05	**96,00**	92,04
CFNet	ResNet-152	84,99	91,41	92,65	91,61
OCRNet	ResNet-152	83,97	92,66	90,27	90,92
U-Net	DenseNet-121	**87,79**	93,15	93,90	**93,23**
FPN	DenseNet-121	**87,05**	**94,11**	92,12	**92,82**
DeepLabv3+	DenseNet-121	85,97	92,02	93,15	92,21
CFNet	DenseNet-121	85,66	91,64	93,11	92,01
OCRNet	DenseNet-121	85,60	92,46	92,28	91,95
U-Net	EfficientNet-B3	**87,10**	**93,95**	92,42	**92,89**
FPN	EfficientNet-B3	86,75	92,64	93,29	92,68
DeepLabv3+	EfficientNet-B3	85,42	88,88	**95,79**	91,85
CFNet	EfficientNet-B3	86,63	91,53	94,36	92,61
OCRNet	EfficientNet-B3	86,48	92,14	93,57	92,52

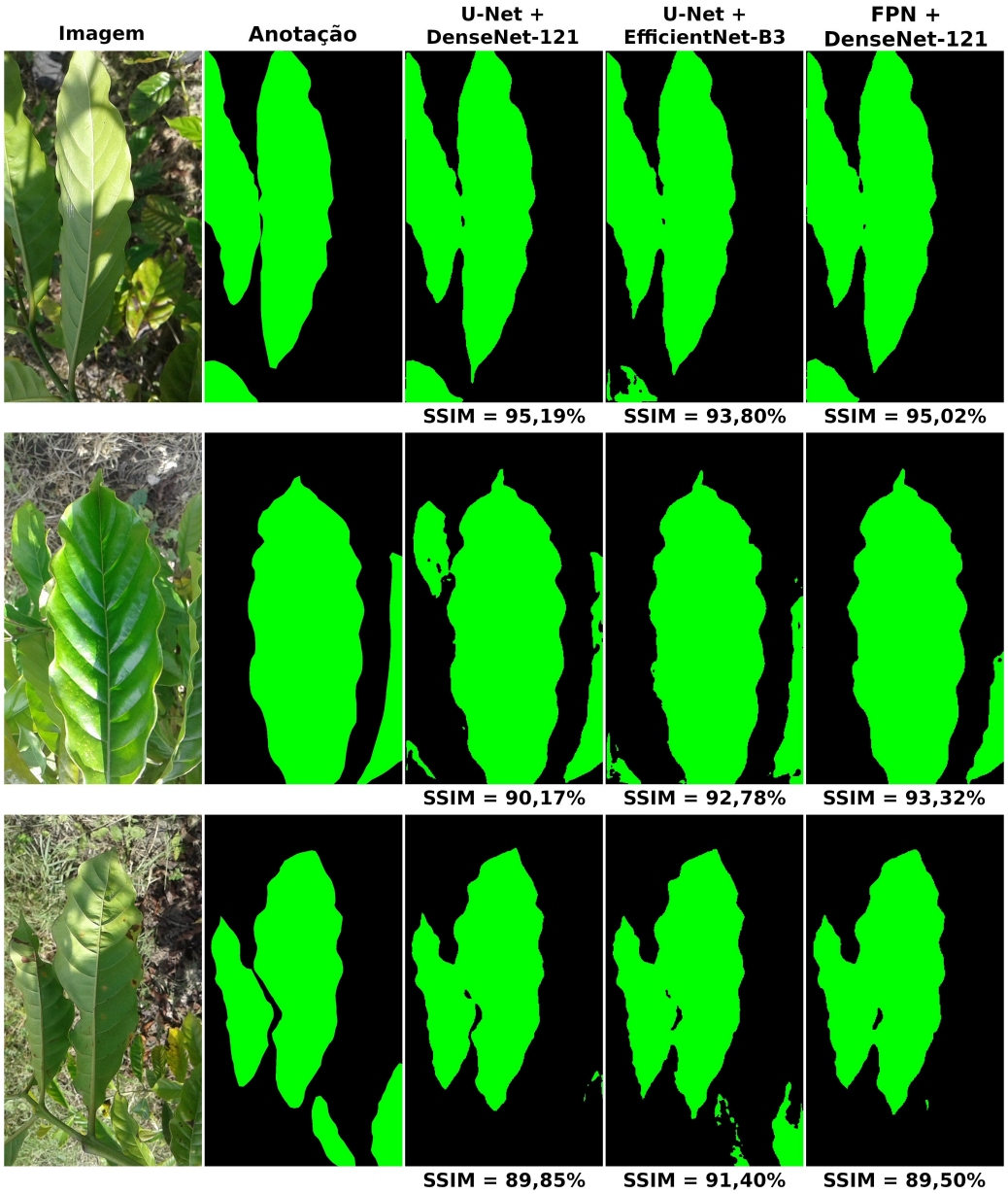
Fonte: Autoria própria (2022).

O modelo que obteve o melhor desempenho geral foi o U-Net treinado com a CNN DenseNet-121 como extratora de características, alcançando tanto o maior IoU (87,79%) quanto o maior *F-score* (93,23%). A segunda colocação também fica com o modelo U-Net, mas agora utilizando a rede EfficientNet-B3 como extratora de características, que garantiu o segundo maior IoU (87,10%) e o segundo maior *F-score* (92,89%), apenas 0,69 pp de IoU e 0,34 pp de *F-score* atrás do primeiro colocado. Fechando as três primeiras posições, o modelo FPN utilizando a CNN pré-treinada DenseNet-121, que conseguiu um IoU de 87,05% e um *F-score* de 92,82%, representando uma diferença de apenas 0,04 pp de IoU e 0,07 pp de *F-score* do segundo colocado.

Possivelmente, o fato dos hiperparâmetros de treinamento utilizados terem sido determinados nos experimentos preliminares com base unicamente na rede de segmentação U-Net pode ter favorecido os resultados do modelo em relação às demais alternativas testadas no escopo deste trabalho, uma vez que modelos com características diferentes podem precisar de hiperparâmetros distintos para uma melhor otimização de seus parâmetros.

A Figura 31 faz uma comparação entre a máscara verdadeira de três imagens do conjunto de testes e a inferência dos três melhores modelos citados. Também são apresentados os índices de similaridade estrutural (SSIM) de cada máscara gerada pelo modelo em relação a anotação de referência. Quanto mais alto o valor do SSIM, maior a semelhança entre a inferência do modelo e o resultado esperado.

Figura 31 – Inferência dos três melhores modelos em três imagens do conjunto de testes



Fonte: Autoria própria (2022).

Os modelos conseguiram encontrar e segmentar as folhas principais presentes nas imagens, especialmente quando estavam corretamente focadas e apresentavam uma separação clara do plano de fundo da imagem. Isso pode ser observado na primeira linha da Figura 31, em que as três folhas principais da imagem estão nítidas e o fundo está totalmente desfocado, resultando em inferências com valores altos de SSIM em relação à máscara esperada. No entanto, os modelos demonstraram uma dificuldade maior na segmentação de folhas que não possuem essas características, ou seja, que não apresentam foco bem definido nas folhas ou que não contam com separação clara entre o primeiro e o segundo plano da imagem, como pode ser observado na segunda e terceira linha da Figura 31. Nesses casos, o SSIM das máscaras geradas cai devido às diferenças observadas entre as máscaras esperadas e às preditas pelos modelos de segmentação.

Esse foi justamente o principal problema observado na inferência gerada pelos modelos: a presença de folhas extras ou a ausência de folhas marcadas em comparação com a máscara verdadeira. A abordagem geral adotada para a segmentação manual consistiu na marcação de todas as folhas focadas do primeiro plano da imagem. No entanto, muitas dessas folhas não podiam ser segmentadas de forma totalmente precisa, já que estavam justamente no limiar entre o primeiro e o segundo plano. Esse fato pode ter tornado a marcação subjetiva e resultado em ambiguidades que confundem os modelos treinados, dificultando o aprendizado e prejudicando o desempenho final.

Um desempenho interessante pode ser observado a partir da análise dos resultados obtidos pelo modelo DeepLabv3+, principalmente nos casos em que utilizou as CNNs ResNet-152, EfficientNet-B3 e ResNet-50 como extratoras de características: esses treinamentos apresentaram as maiores coberturas dentre todos os resultados obtidos (96%, 95,79% e 95%, respectivamente), o que significa que se tornaram melhores do que as outras redes na tarefa de encontrar todas as folhas anotadas nas imagens. Por outro lado, a cobertura acima da média apresentada nesses casos teve um preço: ambos apresentaram os piores valores de precisão (89,05%, 88,88% e 89,81%, respectivamente), o que indica que não se saíram tão bem em segmentar apenas os *pixels* que realmente pertencem às folhas.

Quanto a CNN extratora de características, o destaque fica com o modelo DenseNet-121, que apareceu duas vezes entre os três melhores resultados obtidos nos treinamentos realizados (com U-Net e FPN). Outra CNN presente entre os melhores resultados foi a EfficientNet-B3, tendo inclusive se mantido mais consistente em seus resultados que as demais CNNs utilizadas. Por outro lado, a CNN ResNet-152 foi a que obteve o pior desempenho geral, apresentando resultados inferiores para quase todos os modelos de segmentação testados. Para alguns modelos de segmentação, como é o caso do DeepLabv3+, não é possível perceber variações significativas de desempenho entre as diferentes CNNs utilizadas como extratoras de características, bem diferente do que ocorre com modelos como U-Net ou FPN, que tiveram variações consideráveis de desempenho de acordo com a CNN utilizada.

A Tabela 10 apresenta dados relacionados com o tamanho de lote, número de épocas e tempo de treinamento de cada experimento realizado, com destaque em negrito para os três menores tempos de treinamento. Além da resolução das imagens utilizadas durante o treinamento, o tamanho do lote é influenciado pela quantidade de parâmetros do modelo de segmentação e da CNN extratora de características, de forma que modelos computacionalmente menos custosos permitem o uso de um tamanho de lote maior. Nesse quesito, apenas o modelo FPN apresentou desvantagem, visto que só permitiu o treinamento com metade do tamanho de lote dos demais modelos para todas as CNNs pré-treinadas utilizadas.

Tabela 10 – Informações de treinamento dos modelos de segmentação

Modelo	Extrator de características	Tamanho do lote de treinamento	Quantidade de épocas treinadas	Tempo de treinamento (h:m:s)
U-Net	ResNet-50	16	96	**05:33:32**
FPN	ResNet-50	8	185	16:55:52
DeepLabv3+	ResNet-50	16	300	18:33:00
CFNet	ResNet-50	16	300	17:07:06
OCRNet	ResNet-50	16	287	16:10:02
U-Net	ResNet-152	8	83	**06:59:45**
FPN	ResNet-152	4	208	23:52:18
DeepLabv3+	ResNet-152	8	270	18:47:47
CFNet	ResNet-152	8	254	20:28:30
OCRNet	ResNet-152	8	244	16:45:36
U-Net	DenseNet-121	16	127	07:26:39
FPN	DenseNet-121	8	108	07:39:12
DeepLabv3+	DenseNet-121	16	381	21:57:08
CFNet	DenseNet-121	16	365	21:11:57
OCRNet	DenseNet-121	16	354	22:18:09
U-Net	EfficientNet-B3	8	77	**05:04:03**
FPN	EfficientNet-B3	4	78	07:49:48
DeepLabv3+	EfficientNet-B3	8	235	16:16:45
CFNet	EfficientNet-B3	8	282	19:10:48
OCRNet	EfficientNet-B3	8	331	19:55:31

Fonte: Autoria própria (2022).

O modelo que apresentou o menor tempo de treinamento foi o U-Net, tendo convergido mais rapidamente do que os outros modelos em todos os testes feitos, indiferente da CNN pré-treinada utilizada como extratora de características. Além disso, o modelo também se destaca quanto ao baixo consumo computacional necessário para seu treinamento em relação aos demais modelos testados. Essas características observadas do modelo U-Net podem estar relacionadas com sua estrutura relativamente mais simples do que os demais modelos, permitindo uma convergência mais rápida utilizando menos memória.

Nesse quesito, o modelo FPN foi o mais inconstante, visto que teve uma grande variação no tempo de treinamento e no número de épocas necessárias para a convergência conforme a CNN pré-treinada utilizada como extratora de características, principalmente nos experimentos utilizando as redes da família ResNet (ResNet-50 e ResNet-152). O fato da rede ser computa-

cionalmente mais custosa obriga a redução do tamanho de lote utilizado no treinamento, o que também pode contribuir para a instabilidade apresentada.

Quanto às CNNs pré-treinadas utilizadas, o destaque nesse quesito vai para o modelo EfficientNet-B3, que além de demonstrar um desempenho consistente, também foi a CNN que apresentou os menores tempos de treinamento para a maioria dos modelos de segmentação utilizados, mostrando-se como uma opção promissora para a extração de características em tarefas de segmentação. Outro modelo que também apresentou tempos de treinamento consistentes foi o ResNet-50, podendo ser uma boa opção quando os recursos computacionais disponíveis para o treinamento são mais limitados.

Embora existam diferenças significativas entre as bases de dados utilizadas e abordagens metodológicas seguidas, alguns resultados observados nos experimentos desenvolvidos no escopo deste trabalho são comuns aos obtidos nos trabalhos correlatos mencionados na Seção 2.10. Nos testes conduzidos por Esgario *et al.* (2021), assim como nos experimentos de Gonçalves *et al.* (2021) e de Ngugi, Abelwahab e Abo-Zahhad (2020), apesar de diferentes modelos de aprendizado profundo terem sido utilizados para a segmentação semântica de folhas, a rede U-Net sempre apareceu entre os modelos com melhores desempenhos. Além disso, o modelo FPN também se destacou nos experimentos de Gonçalves *et al.* (2021), o que condiz com os resultados obtidos neste trabalho.

Utilizando a rede U-Net, Esgario *et al.* (2021) obteve um IoU de 94,85%, valor 7,06 pp mais alto do que o maior IoU obtido nos experimentos realizados neste trabalho (87,79%). Entretanto, é preciso levar em consideração as diferenças existentes na tarefa de segmentação dos dois trabalhos: Esgario *et al.* (2021) segmentou semanticamente imagens de folhas de café tiradas em condições de laboratório e com uma única folha por imagem; por outro lado, a proposta de segmentação utilizada no escopo deste trabalho envolve a segmentação de imagens tiradas em condições reais de campo com múltiplas folhas por imagem, o que representa um grau de complexidade maior e justifica as diferenças observadas no desempenho obtido.

5 CONSIDERAÇÕES FINAIS

Este capítulo se destina a apresentação das conclusões e contribuições deste trabalho. Também serão discutidos possíveis trabalhos futuros, que podem ser desenvolvidos visando melhorar os resultados obtidos ou aplicá-los em outros problemas da área.

5.1 Conclusão

Este trabalho teve por objetivo a aplicação de técnicas de aprendizado profundo para a segmentação semântica de imagens de folhas de café. As técnicas testadas consistem em redes neurais profundas que se destacam na literatura por conta de seus bons resultados na tarefa de segmentação de objetos na cena, incluindo os modelos U-Net, FPN, DeepLabv3+, CFNet e OCRNet. Cada modelo foi treinado utilizando 4 diferentes CNNs pré-treinadas como extratoras de características: ResNet-50, ResNet-152, DenseNet-121 e EfficientNet-B3. Os Experimentos utilizaram imagens de três bases de dados públicas da cultura de interesse, que precisaram passar por processo de anotação. Visando expandir ainda mais a quantidade de exemplos disponíveis para o treinamento, foram utilizados métodos de aumento de dados que resultaram na geração de um conjunto de imagens sintéticas.

Todos os modelos treinados apresentaram IoU acima dos 83% e *F-score* acima dos 90%. As inferências analisadas demonstram que as redes aprenderam corretamente a identificar as folhas presentes nas imagens e separá-las dos elementos do fundo. A principal dificuldade observada envolve a correta distinção entre quais folhas realmente devem ser segmentadas, dado que a abordagem adotada durante a anotação excluiu folhas desfocadas ou que se encontravam no fundo da imagem. No entanto, isso pode ter levado a ambiguidades que prejudicaram o aprendizado dos modelos, visto os principais erros observados envolvem justamente a identificação de folhas que não foram anotadas ou a ausência de folhas que foram marcadas na anotação.

Dentre os resultados obtidos, o modelo U-Net foi o que apresentou o melhor desempenho, tanto quando foi treinado utilizando a CNN pré-treinada DenseNet-121 (87,79% de IoU e 93,23% de *F-score*), quanto no treinamento usando a CNN pré-treinada EfficientNet-B3 (87,10% de IoU e 92,89% de *F-score*). Além disso, o modelo U-Net também precisou de menos épocas de treinamento e de um tempo menor para convergir em todos os testes feitos. O modelo FPN, embora computacionalmente mais custoso que as demais opções testadas no escopo deste trabalho, também apresentou ótimos resultados e o terceiro melhor desempenho (87,05% de IoU e 92,82% de *F-score*), podendo ser uma ótima opção em casos que os recursos computacionais disponíveis para o treinamento não são uma limitação. No que diz respeito às CNNs pré-treinadas utilizadas como extratoras de características, o destaque fica com as redes DenseNet-121 e EfficientNet-B3, sendo as que mais apareceram entre os melhores resultados obtidos.

É válido destacar que as diferenças de desempenho entre os modelos com os melhores resultados dos experimentos realizados são mínimas, e se traduzem em apenas alguns poucos detalhes nas máscaras geradas, tais como contornos ou forma das folhas. Só para fins comparativos, se os três maiores valores de IoU (87,79%, 87,10% e 87,05%) fossem escritos em função das 64 imagens do conjunto de teste utilizado nos experimentos, teríamos valores de 56,18, 55,74 e 55,71 imagens, uma variação de menos de meia imagem. Uma diferença maior fica evidente analisando-se o menor IoU obtido (83,97%), que corresponde a 53,74 imagens, uma diferença de 2,44 imagens do melhor modelo.

5.2 Trabalhos futuros

Por conta do tempo disponível para a conclusão do trabalho, não foi possível realizar a anotação de cerca de 800 imagens do *dataset* criado por Parraga-Alava *et al.* (2019). Espera-se que o acréscimo dessas imagens possa fornecer mais exemplos únicos e promover a melhoria geral do desempenho obtido neste trabalho. Outra alternativa válida é a busca por novos bancos de imagens ou mesmo a construção de um novo *dataset*, visando expandir a quantidade de amostras reais disponíveis para o treinamento dos modelos.

Além disso, como mencionado na Seção 5.1, os critérios adotados durante o processo de anotação das folhas podem ter gerado ambiguidades que atrapalharam o aprendizado dos modelos de aprendizado profundo. Dessa forma, a utilização de uma abordagem de segmentação diferente, como a anotação apenas da folha central ou de todas as folhas identificáveis na imagem, pode contribuir para a diminuição da ambiguidade e promover a melhoria dos resultados obtidos.

Os modelos de segmentação treinados podem ser utilizados para a geração de um conjunto de imagens mais simples de folhas de café, capazes de facilitar o aprendizado e melhorar o desempenho em outras tarefas da área de visão computacional e aprendizado de máquina, como a classificação ou a detecção de objetos. Outra alternativa é a utilização das mesmas técnicas de segmentação para o treinamento de modelos voltados a estimação da severidade dos sintomas foliares causados por pragas ou doenças.

Por fim, outro estudo interessante que pode ser muito útil para a área é a aplicação de outros tipos de segmentação aos dados, principalmente a segmentação de instância. Essa abordagem pode ser utilizada para tarefas que envolvam a contagem de objetos na cena, como folhas ou frutos. Dados como esses podem ser muito relevantes para uma análise da saúde ou produtividade da cultura em questão.

REFERÊNCIAS

AGGARWAL, C. C. **Neural Networks and Deep Learning**. Yorktown Heights, NY: Springer, 2018. ISBN 978-3-319-94463-0.

ALBUMENTATIONS. **Welcome to Albumentations documentation**. 2021. Disponível em: https://albumentations.ai/docs/.

ALOM, M. Z. *et al.* A state-of-the-art survey on deep learning theory and architectures. **Electronics (Switzerland)**, v. 8, 2019. ISSN 20799292.

ARAGÃO, A.; CONTINI, E. **O agro no brasil e no mundo: uma síntese do período de 2000 a 2020**. 2021. Disponível em: http://embrapa.br/documents/10180/62618376/O+AGRO+NO+BRASIL+E+NO+MUNDO.pdf.

ARSENOVIC, M. *et al.* Solving current limitations of deep learning based approaches for plant disease detection. **Symmetry**, v. 11, 2019. ISSN 20738994.

ATIENZA, R. **Advanced Deep Learning with TensorFlow 2 and Keras**. 2. ed. Birmingham, UK: Packt Publishing, 2020. 512 p. ISBN 9781838821654.

BARBEDO, J. G. A. A review on the main challenges in automatic plant disease identification based on visible range images. **Biosystems Engineering**, v. 144, 2016. ISSN 15375110.

BARBEDO, J. G. A. Plant disease identification from individual lesions and spots using deep learning. **Biosystems Engineering**, v. 180, 2019. ISSN 15375110.

BARBEDO, J. G. A. *et al.* Annotated plant pathology databases for image-based detection and recognition of diseases. **IEEE Latin America Transactions**, v. 16, p. 1749–1757, 6 2018. ISSN 1548-0992.

BARBEDO, J. G. A.; KOENIGKAN, L. V.; SANTOS, T. T. Identifying multiple plant diseases using digital image processing. **Biosystems Engineering**, v. 147, 2016. ISSN 15375110. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/handle/1/14513.

CAMPO & NEGÓCIO. **Anuário do café**. 2021. Disponível em: https://www.yumpu.com/pt/document/read/65476088/.

CARVALHO, C. N. de *et al.* Coffea arabica l.: potencialidades e ações medicinais. **Revista Intertox de Toxicologia, Risco Ambiental e Sociedade**, v. 11, 2018. ISSN 1984-3577.

CARVALHO, V. L.; CHALFOUN, S. M.; CUNHA, R. L. da. **Doenças do cafeeiro: diagnose e controle**. Belo Horizonte, MG: EPAMIG, 2013. ISSN 0101-062X.

CHEN, L. C. *et al.* Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE Computer Society, v. 40, p. 834–848, 4 2018. ISSN 01628828.

CHEN, L.-C. *et al.* Encoder-decoder with atrous separable convolution for semantic image segmentation. In: . [*s.n.*], 2018. Disponível em: http://arxiv.org/abs/1802.02611.

COCATO, L.; D'ARC, J. C. **Diferenças das espécies Coffea arabica e Coffea canephora**. 2020. Disponível em: https://rehagro.com.br/blog/diferencas-das-especies-coffea-arabica-e-coffea-canephora-2/.

CVAT. **Getting started**. 2021. Disponível em: https://openvinotoolkit.github.io/cvat/docs/getting_started/.

DATA SCIENCE ACADEMY. **Deep Learning Book**. 2021. Disponível em: https://www.deeplearningbook.com.br.

DAVIS, A. P. F. *et al.* Growing coffee: Psilanthus (rubiaceae) subsumed on the basis of molecular and morphological data; implications for the size, morphology, distribution and evolutionary history of coffea. **Botanical Journal of the Linnean Society**, v. 167, p. 357–377, 2011.

DOSSELMANN, R.; YANG, X. D. A comprehensive assessment of the structural similarity index. **Signal, Image and Video Processing**, Springer London, v. 5, p. 81–91, 3 2011. ISSN 18631711.

DURÁN, C. A. A. *et al.* Café: Aspectos gerais e seu aproveitamento para além da bebida coffee: General aspects and its use beyond drink. **Revista Virtual de Química**, 2016.

ESGARIO, J. *et al.* An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning. **Information Processing in Agriculture**, 2021.

ESGARIO, J. G.; KROHLING, R. A.; VENTURA, J. A. Deep learning for classification and severity estimation of coffee leaf biotic stress. **Computers and Electronics in Agriculture**, Elsevier B.V., v. 169, 2 2020. ISSN 01681699.

FACELI, K. *et al.* **Inteligência Artificial: uma abordagem de aprendizado de máquina**. Rio de Janeiro, RJ: LTC, 2011. ISBN 978-85-216-1880-5.

FAWAZ, H. I. *et al.* Data augmentation using synthetic data for time series classification with deep residual networks. In: . [*S.l.*]: arXiv, 2018.

FERRÃO, R. G. *et al.* **Café Conilon**. 2. ed. Vitória, ES: Incaper, 2017. ISBN 978-85-89274-26-5.

GARCIA-GARCIA, A. *et al.* A review on deep learning techniques applied to semantic segmentation. **CoRR**, abs/1704.06857, 2017. Disponível em: http://arxiv.org/abs/1704.06857.

GHOSH, S. *et al.* Understanding deep learning techniques for image segmentation. **ACM Computing Surveys**, Association for Computing Machinery, v. 52, 8 2019. ISSN 15577341.

GONÇALVES, J. P. *et al.* Deep learning architectures for semantic segmentation and automatic estimation of severity of foliar symptoms caused by diseases or pests. **Biosystems Engineering**, Academic Press, v. 210, p. 129–142, 10 2021. ISSN 15375110.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. 4. ed. New York: Pearson Education, 2018. ISBN 9780133356724.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. **Deep Learning**. Cambridge, Massachusetts: Massachusetts Institute of Technology, 2016. ISBN 9780262035613.

HAYKIN, S. **Redes neurais: princípios e prática**. 2. ed. Hamilton, Ontario: Bookman, 2005. ISBN 0-13-273350-1.

HE, K. *et al.* Deep residual learning for image recognition. 12 2015. Disponível em: http://arxiv.org/abs/1512.03385.

HEATON, J. **Artificial intelligence for humans**. 1. ed. Clarkston, Georgia: Heaton Research, 2015. v. 3. ISBN 978-1505714340.

HUANG, G. *et al.* Densely connected convolutional networks. 8 2016. Disponível em: http://arxiv.org/abs/1608.06993.

HUELLMANN, T. **What is precision vs recall in machine learning?** 2022. Disponível em: https://levity.ai/blog/precision-vs-recall.

KATAOKA, T. *et al.* Crop growth estimation system using machine vision. **IEEE/ASME International Conference on Advanced Intelligent Mechatronics, AIM**, v. 2, 2003.

KEZMANN, J.-M. **Tensorflow Advanced Segmentation Models**. GitHub, 2020. Disponível em: https://github.com/JanMarcelKezmann/TensorFlow-Advanced-Segmentation-Models.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. 12 2014. Disponível em: http://arxiv.org/abs/1412.6980.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, Nature Publishing Group, v. 521, p. 436–444, 5 2015. ISSN 14764687.

LIN, T.-Y. *et al.* Feature pyramid networks for object detection. In: . [*S.l.*]: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

LIN, T.-Y. *et al.* Focal loss for dense object detection. 8 2017. Disponível em: http://arxiv.org/abs/1708.02002.

MARTINS, A. L. **História do Café**. 2. ed. São Paulo, SP: Editora Contexto, 2008. v. 1. ISBN 978-85-7244-377-7.

MATPLOTLIB. **Matplotlib: Visualization with Python**. 2021. Disponível em: https://matplotlib.org/.

MELO, C. M. de *et al.* Next-generation deep learning based on simulators and synthetic data. **Trends in Cognitive Sciences**, Elsevier Ltd, v. 26, p. 174–187, 2 2022. ISSN 1879307X.

MESQUITA, C. M. de *et al.* **Manual do café: distúrbios fisiológicos, pragas e doenças do cafeeiro**. Belo Horizonte, MG: Emater, 2016.

MINAEE, S. *et al.* Image segmentation using deep learning: A survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, IEEE Computer Society, 2021. ISSN 19393539.

MITCHELL, T. M. T. M. **Machine Learning**. 1. ed. [*S.l.*: *s.n.*], 1997. 432 p. ISBN 0070428077.

NGUGI, L. C.; ABELWAHAB, M.; ABO-ZAHHAD, M. Tomato leaf segmentation algorithms for mobile phone applications using deep learning. **Computers and Electronics in Agriculture**, Elsevier B.V., v. 178, 11 2020. ISSN 01681699.

NIKOLENKO, S. I. **Synthetic Data for Deep Learning**. Springer Optimization and Its Applications, 2021. Disponível em: http://www.springer.com/series/7393.

OLIVEIRA, C. M. *et al.* Crop losses and the economic impact of insect pests on brazilian agriculture. **Crop Protection**, v. 56, 2014. ISSN 02612194.

ORGANIZAÇÃO INTERNACIONAL DO CAFÉ. **Annual review: coffee year 2019/2020**. 2020. Disponível em: http://www.ico.org/documents/cy2020-21/annual-review-2019-2020-e.pdf.

OUAKNINE, A. Review of deep learning algorithms for image semantic segmentation. **Valeo**, 2019.

PAPERS WITH CODE. **Image Classification on ImageNet**. 2022. Disponível em: https://paperswithcode.com/sota/image-classification-on-imagenet.

PARRAGA-ALAVA, J. *et al.* Rocole: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition. **Mendeley Data**, v. 2, 2019.

PATKI, N.; WEDGE, R.; VEERAMACHANENI, K. The synthetic data vault. In: . [*S.l.*]: Institute of Electrical and Electronics Engineers Inc., 2016. p. 399–410. ISBN 9781509052066.

PATRAWALA, V. **Create A Synthetic Image Dataset - The "What", The "Why" and The "How"**. 2020. Disponível em: https://towardsdatascience.com/create-a-synthetic-image-dataset-the-what-the-why-and-the-how-f820e6b6f718.

PEDRINI, H.; SCHWARTZ, W. R. **Análise de imagens digitais: princípios, algoritmos e aplicações**. São Paulo: Cengage Learning, 2007.

PIRES, W. O. **Reconhecimento de espécies florestais pela folha, utilizando redes neurais convolucionais**. 2018. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2018.

PONTE, S. The latte revolution? regulation, markets and consumption in the global coffee chain. **World Development**, v. 30, p. 1099–1122, 2002. Disponível em: www.elsevier.com/locate/worlddev.

PYTHON BRASIL. **Python para quem está começando**. 2021. Disponível em: https://python.org.br/introducao/.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: . [*S.l.*]: Springer Verlag, 2015. v. 9351, p. 234–241. ISBN 9783319245737. ISSN 16113349.

SCIKIT-IMAGE. **Structural similarity index**. 2021. Disponível em: https://scikit-image.org/docs/stable/auto_examples/transform/plot_ssim.html.

SHALEV-SHWARTZ, S.; BEN-DAVID, S. **Understanding Machine Learning**. Cambridge, Massachusetts: Cambridge University, 2014. ISBN 978-1-107-05713-5.

SHARMA, P.; BERWAL, Y. P. S.; GHAI, W. Performance analysis of deep learning cnn models for disease detection in plants using image segmentation. **Information Processing in Agriculture**, China Agricultural University, v. 7, p. 566–574, 12 2020. ISSN 22143173.

SILVA, L. B.; CARNEIRO Álvaro L. C.; FAULIN, M. S. A. R. **rust (Hemileia vastatrix) and leaf miner (Leucoptera coffeella) in coffee crop (Coffea arabica)**. [*S.l.*]: Mendeley Data, 2020.

SOLOMON, C.; BRECKON, T. **Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab**. 1. ed. Chichester: Wiley-Blackwell, 2011. ISBN 9780470689776.

SOUZA, F. de F. *et al.* Características das principais variedades de café cultivadas em rondônia. **Embrapa Rondônia**, 4 2004. ISSN 0103-9865. Disponível em: www.cpafro.embrapa.br.

TAN, M.; LE, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. 5 2019. Disponível em: http://arxiv.org/abs/1905.11946.

TASKESEN, E. **Undouble's documentation!** 2020. Disponível em: https://erdogant.github.io/undouble/pages/html/index.html.

TAYLOR, M. **Neural Networks A Visual Introduction for Beginners**. [*S.l.*]: Blue Windmill Media, 2017. ISBN 9781549869136.

TENSORFLOW. **Introdução ao TensorFlow**. 2021. Disponível em: https://www.tensorflow.org/learn.

TREBESCHI, S. *et al.* Deep learning for fully-automated localization and segmentation of rectal cancer on multiparametric mr. **Scientific Reports**, Nature Publishing Group, v. 7, 12 2017. ISSN 20452322.

TREML, M. *et al.* Speeding up semantic segmentation for autonomous driving. **29th Conference on Neural Information Processing Systems (NIPS 2016)**, 2016.

VASILEV, I. *et al.* **Python deep learning: exploring deep learning techniques and neural network architectures with PyTorch, Keras, and TensorFlow**. 2. ed. Birmingham: Packt Publishing, 2019. ISBN 978-1-78934-846-0.

WANG, Z. *et al.* Image quality assessment: From error visibility to structural similarity. **IEEE Transactions on Image Processing**, v. 13, p. 600–612, 4 2004. ISSN 10577149.

XIONG, Y. *et al.* Identification of cash crop diseases using automatic image segmentation algorithm and deep learning with expanded dataset. **Computers and Electronics in Agriculture**, v. 177, 2020. ISSN 01681699.

XU, L. *et al.* Leaf instance segmentation and counting based on deep object detection and segmentation networks. **Proceedings - 2018 Joint 10th International Conference on Soft Computing and Intelligent Systems and 19th International Symposium on Advanced Intelligent Systems, SCIS-ISIS 2018**, 2018.

YAKUBOVSKIY, P. **Segmentation Models**. GitHub, 2019. Disponível em: https://github.com/qubvel/segmentation_models.

YUAN, Y. *et al.* Segmentation transformer: Object-contextual representations for semantic segmentation. **ECCV 2020**, 2020. Disponível em: https://git.io/HRNet.OCR.

ZHANG, H. *et al.* Co-occurrent features in semantic segmentation. In: . [*s.n.*], 2019. Disponível em: http://hangzh.com/.