Detecção de linhas de plantio em plantações de cana-de-açúcar utilizando *deep learning*

João Batista Ribeiro

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2023

João Batista Ribeiro

Detecção de linhas de plantio em plantações de cana-de-açúcar utilizando *deep learning*

Dissertação de mestrado apresentada ao Programa de Pós-graduação da Faculdade de Computação da Universidade Federal de Uberlândia como parte dos requisitos para a obtenção do título de Mestre em Ciência da Computação.

Área de concentração: Ciência da Computação

Orientador: Prof. Dr. André Ricardo Backes
Coorientador: Prof. Dr. Mauricio Cunha Escarpinati

Uberlândia
2023

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

R354d
2023

Ribeiro, João Batista, 1991-
Detecção de linhas de plantio em plantações de cana-de-açúcar utilizando *deep learning* [recurso eletrônico] / João Batista Ribeiro. - 2023.

Orientador: André Ricardo Backes.
Coorientador: Mauricio Cunha Escarpinati.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Ciência da Computação.
Modo de acesso: Internet.
Disponível em: http://doi.org/10.14393/ufu.di.2023.8085
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Backes, André Ricardo, 1981-, (Orient.). II. Escarpinati, Mauricio Cunha, 1976-, (Coorient.). III. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

CDU: 681.3

André Carlos Francisco
Bibliotecário - CRB-6/3408

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Ciência da Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação de Mestrado 08/2023, PPGCO				
Data:	16 de agosto de 2023	Hora de início:	09:00	Hora de encerramento:	10:45
Matrícula do Discente:	12112CCP016				
Nome do Discente:	João Batista Ribeiro				
Título do Trabalho:	Detecção de linhas de plantio em plantações de cana-de-açúcar utilizando deep learning				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Ciência de Dados				
Projeto de Pesquisa de vinculação:	-				

Reuniu-se, por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Maurício Cunha Escarpinati e Henrique Coelho Fernandes - FACOM/UFU, João Fernando Mari - UFV e André Ricardo Backes - FACOM/UFU, orientador do candidato.

Os examinadores participaram desde as seguintes localidades: João Fernando Mari - Rio Paranaíba/MG, Maurício Cunha Escarpinati, Henrique Coelho Fernande e André Ricardo Backes - Uberlândia/MG. O discente participou da cidade de Uberlândia/MG.

Iniciando os trabalhos a presidente da mesa, Prof. Dr. André Ricardo Backes, apresentou a Comissão Examinadora e o candidato, agradeceu a presença do público, e concedeu ao Discente a palavra para a exposição do seu trabalho. A duração da apresentação do Discente e o tempo de arguição e resposta foram conforme as normas do Programa.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o candidato. Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato:

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Mauricio Cunha Escarpinati**, **Professor(a) do Magistério Superior**, em 18/08/2023, às 14:24, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Henrique Coelho Fernandes**, **Professor(a) do Magistério Superior**, em 18/08/2023, às 14:37, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **André Ricardo Backes**, **Usuário Externo**, em 18/08/2023, às 14:58, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **João Fernando Mari**, **Usuário Externo**, em 18/08/2023, às 17:05, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4743125** e o código CRC **A30E0299**.

Referência: Processo nº 23117.059267/2023-04 SEI nº 4743125

Dedico este projeto aos meus queridos avós maternos Maria Abadia Xavier (in memoriam) e Jaci Ribeiro Xavier (in memoriam), cuja presença foi essencial na minha vida. Além de sempre me motivarem e acreditarem no meu potencial.

Agradecimentos

Primeiramente, ao meu orientador, Prof. Dr. André Ricardo Backes, pelos grandes ensinamentos, críticas, sugestões e ajuda nos mais variados problemas (principalmente nos experimentos). Ao meu coorientador, Prof. Dr. Mauricio Cunha Escarpinati, pelas críticas, ensinamentos e oportunidades que me foram dadas. Ambos, além de serem ótimos pesquisadores e cientistas, são ótimas pessoas, me motivando e apoiando durante todo projeto.

À minha querida namorada, Mariana Ishhanuhi da Silva Sismanoglu, por todo apoio, carinho, motivação, paciência e por conseguir me aturar nos momentos mais complicados. Além de motivar e incentivar a me desenvolver a cada dia mais. Diante da vastidão do tempo e da imensidão do universo, é um prazer para mim dividir um planeta e uma época contigo, assim como disse Carl Sagan.

Ao meu grande amigo, Leandro Henrique Furtado Pinto Silva, por me motivar a iniciar nessa jornada, me tirar várias dúvidas e me dar várias sugestões e conselhos. Além de me apresentar ao Prof. André, me fazendo a propaganda dele ser um ótimo orientador, paciente e compreendido, o que de fato é verdade.

Aos órgãos de fomento por motivar, financiar e tornar possível a pesquisa nas mais variadas áreas. Em especial à Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG) que apoiou este projeto.

À minha mãe, Luciane Aparecida Ribeiro, que do jeito dela tem me motivado e apoiado a estudar e lutar por um futuro melhor.

À empresa Sensix Inovações em Drones LTDA por ter disponibilizado parte das imagens utilizadas nos experimentos.

À banca, composta pelo Prof. Dr. Henrique Coelho Fernandes e pelo Prof. Dr. João Fernando Mari, pelas valiosas contribuições, correções e sugestões, as quais desempenharam um papel crucial no aprimoramento deste projeto.

A todos professores, técnicos e demais empregados da Faculdade de Computação (FACOM) da Universidade Federal de Uberlândia (UFU) que me ajudaram diretamente ou indiretamente, em especial ao secretário Erisvaldo Araújo Fialho, que sempre foi muito

prestativo ao tirar minhas dúvidas.

No mais, a todas as pessoas que me ajudaram de uma maneira ou de outra, deixo aqui o meu Muito Obrigado e Viva a Ciência!

"Ever tried. Ever failed. No matter. Try again. Fail again. Fail better."
(Samuel Beckett)

Resumo

O rápido crescimento populacional tem impulsionado a demanda por alimentos e a utilização sustentável dos recursos naturais. Nesse contexto, a agricultura aliada à tecnologia, denominada de Agricultura de Precisão (AP), busca suprir essa demanda utilizando os recursos sob medida com base nas informações coletadas. As imagens utilizadas na AP têm variadas fontes, por exemplo, câmeras acopladas em Veículos Aéreos Não Tripulados (VANTs). Uma das principais aplicações da AP é a detecção das linhas de plantio, principalmente porque esta é uma etapa importante para outras aplicações da AP, como detecção de ervas daninhas, mapeamento e previsão de produção de safra, detecção de falhas. Um dos cenários de grande utilização da AP no Brasil está no cultivo de cana-de-açúcar, motivando pesquisadores e empresas a desenvolverem soluções na área. Apesar de existirem na literatura muitos trabalhos para detectar linhas de plantio, a maioria deles é para outras culturas, como milho e beterraba, ou focados em linhas retas. Considerando o cenário de grande utilização dos VANTs para obtenção de imagens para a AP e a grande importância da detecção das linhas de plantio, este projeto analisou diferentes modelos de *Deep Learning* (DL) para segmentação automática em imagens de VANTs de plantações de cana-de-açúcar com variados estágios de crescimento. Dentre os modelos, a U-Net obteve os melhores resultados, com 0,90 ou mais de Coeficiente de *Dice* (CD) para quase todos cenários. Também foi analisado a utilização de Índices de Vegetação (IVs) e Operações Morfológicas de modo a otimizar a detecção das linhas de plantio. Com base nos resultados, são apresentadas algumas recomendações de utilização da U-Net e dos IVs para obter uma maior precisão na segmentação das imagens de VANTs de plantações de cana-de-açúcar.

Palavras-chave: Linhas de Plantio, *Deep Learning*, Agricultura de Precisão, VANTs.

Abstract

Rapid population growth has driven demand for food and the sustainable use of natural resources. In this context, agriculture allied to technology, called Precision Agriculture (PA), seeks to meet this demand using tailored resources based on the information collected. The images used in the PA have different sources, e.g., cameras attached to Unmanned Aerial Vehicles (UAVs). One of the main applications of PA is the detection of planting lines, mainly because it is an important step for other PA applications, e.g., weed detection, crop production mapping and forecasting, fault detection. One of the scenarios of great use of PA in Brazil is in the cultivation of sugarcane, motivating researchers and companies to develop solutions in the area. Although there are many works in the literature to detect planting lines, most of them are for other crops, e.g., maize and beet, or focused on straight lines. Considering the scenario of great use of UAVs to obtain images for PA and the great importance of detecting planting lines, this project analyzed different Deep Learning (DL) models for automatic segmentation in UAV images of sugarcane plantations with varying stages of growth. Among the models, U-Net achieved the best results, with 0.90 or more Dice Coefficient (DC) for almost all scenarios. The use of Vegetation Indexes (VIs) and Morphological Operations was also analyzed in order to optimize the detection of planting lines. Based on the results, some recommendations are presented for using U-Net and VIs to obtain greater precision in the segmentation of UAVs images of sugarcane plantations.

Keywords: Planting lines, Deep Learning, Precision Agriculture, UAVs.

Lista de ilustrações

Lista de tabelas

Lista de siglas

AG Algoritmo Genético

AP Agricultura de Precisão

CD Coeficiente de *Dice*

CJ Coeficiente de Jaccard

CNN *Convolutional Neural Network*

DL *Deep Learning*

ExG *Excess Green index*

FCN *Fully Convolutional Netwok*

GPS Sistema de Posicionamento Global

GSD *Ground Sample Distance*

IV Índice de Vegetação

IoU *Intersection over Union*

KNN *K-nearest neighbors*

NDVI *Normalized Difference Vegetation Index*

NIR *Near Infrared*

PDI Processamento Digital de Imagens

RGB *Red*, *Blue* e *Green*

SVM *Support Vector Machine*

TH Transformada de Hough

TR Transformada de Radon

VANT Veículo Aéreo Não Tripulado

Sumário

Capítulo 1

Introdução

O rápido crescimento populacional, principalmente no último século, atingindo 8 bilhões no final de 2022 (WALSH, 2023), tem impulsionado a demanda por alimentos e a utilização inteligente e sustentável dos recursos naturais. Nesse contexto, a agricultura aliada à tecnologia, denominada Agricultura de Precisão (AP), busca suprir essa demanda utilizando os recursos sob medida com base nas informações coletadas. A AP engloba uma série de técnicas diferentes, como a análise espacial da área plantada, informações do solo e das plantas, o que permite aos produtores planejar e monitorar suas plantações (BLASCH et al., 2020).

A AP se tornou possível graças ao desenvolvimento e avanços de diferentes tecnologias, como o Sistema de Posicionamento Global (GPS), imagens de satélites, o desenvolvimento de novas técnicas de Processamento Digital de Imagens (PDI) e Visão Computacional, sensoriamento remoto, dentre outras. Isso possibilitou o desenvolvimento de metodologias, técnicas e programas que são aplicados nas várias etapas da agricultura, desde análise e preparação do solo (para avaliar a escassez de determinado nutriente em uma região) até a utilização de veículos autônomos para fazer a pulverização de defensivos (com a quantidade específica para cada parte do talhão) e na colheita seguindo as linhas de plantio. Muitos dos avanços na AP são fortemente dependentes das tecnologias de Processamento Digital de Imagens (BOLFE et al., 2020).

As imagens utilizadas na AP têm variadas fontes, e.g., câmeras acopladas em Veículo Aéreo Não Tripulado (VANT) e satélites, dependendo da aplicação e do *Ground Sample Distance* (GSD) requisitado. GSD refere-se à distância da amostra ao solo, ou seja, quanto cada pixel da imagem obtida representa da região fotografada. Assim, quanto menor for o GSD, mais detalhes a imagem terá da região analisada. Satélites (com GSD em média de metros) conseguem capturar imagens de grandes regiões mais facilmente que os VANTs (GSD em média de centímetros), que necessitam de planos de voo para abranger toda região e do processo de mosaicagem para combinar as imagens obtidas (MESSINA et al., 2020).

O custo das imagens de satélites depende da qualidade das imagens, elas possuem

baixa temporalidade (imagens de uma mesma área em momentos diferentes), baixo nível de detalhes (maior GSD) em relação aos VANTs e são bastante susceptíveis às condições climáticas, como a presença de nuvens. Por outro lado, os VANTs geralmente têm custo baixo para aquisição das imagens, possibilitando suas capturas e recapturas sempre que necessário (alta temporalidade e disponibilidade), com grande nível de detalhes (pequeno GSD) e são pouco afetados pelas nuvens devido a sua altitude de voo (CANDIAGO et al., 2015; DELAVARPOUR et al., 2021).

Uma das principais aplicações da AP é a detecção das linhas de plantio, principalmente porque esta é utilizada como uma etapa importante para outras aplicações da AP, e.g., detecção de ervas daninhas, mapeamento e previsão de produção de safra, detecção de falhas (HASSANEIN; KHEDR; EL-SHEIMY, 2019). Além de ser utilizada pelos veículos autônomos para se guiarem na plantação, o que pode evitar o pisoteamento da cultura (DOHA et al., 2021).

A detecção das linhas de plantio é uma tarefa complexa e com vários desafios. Por exemplo, pode haver a presença de ervas daninhas com assinatura espectral e cor semelhante às das linhas da cultura, dificultando sua detecção. Crescimento irregular da cultura, falhas no plantio, imperfeições e oclusões nas linhas de plantio, manchas de solo e tipos diferentes na mesma plantação, presença de artefatos bloqueando a visão das linhas de cultivo (e.g., árvores), sombras, além de variações nas condições climáticas e de iluminação, são exemplos de variações presentes nas imagens e que podem atrapalhar, ou mesmo comprometer, o processo de detecção (RABAB et al., 2021; DOHA et al., 2021).

Um dos cenários de grande utilização da Agricultura de Precisão no Brasil está no cultivo de cana-de-açúcar, motivando pesquisas tanto na área acadêmica (Souza et al. (2017), Rozo (2019), Bah, Hafiane e Canals (2020), Oliveira (2020)) quanto nas empresas (AERO Engenharia (2017), Ribeiro (2020), InfoRow (2021)) a desenvolverem soluções na área.

O Brasil tem sido historicamente o maior produtor e exportador de cana-de-açúcar do mundo, com aproximadamente 10 milhões de hectares de área plantada, sendo que mundialmente esse valor é de pouco mais de 26 milhões de hectares (OGUNSOLA; ZANCANER; SEED, 2021; WALTON; MANSA; SCHMITT, 2022; FAOSTAT, 2022). O seu cultivo é de grande importância para a economia brasileira devido às suas diversas aplicações. A cana-de-açúcar pode, por exemplo, ser consumida fresca na alimentação humana ou aplicada na forragem para alimentação animal, produção de açúcar, bebidas, energia e combustíveis. Além disso, seus subprodutos, como o bagaço e a palha, podem ser utilizados na fertilização do solo (OLIVEIRA; MIRANDA; COOKE, 2018).

A cana-de-açúcar é uma cultura semi-perene (seu manejo pode durar anos sem ser necessário um novo plantio), o que adiciona uma motivação a mais para detectar as suas linhas de plantio em relação às outras culturas (e.g., milho e beterraba). Após a primeira colheita, a rebrota das soqueiras (conjunto de raízes e ramos que permanecem no solo

após a colheita da cana-de-açúcar) é colhida anualmente por cerca de 5 a 7 anos, ou mais (RUDORFF et al., 2010). Contudo, isso também adiciona um desafio, pois dependendo do estágio da plantação, folhas secas estão presentes no solo entre as linhas da cultura, dificultando a análise computacional (SILVA, 2020).

Apesar de na literatura existirem muitos trabalhos para detectar linhas de plantio, a maioria deles é para outras culturas (como milho e beterraba) ou focados em linhas retas (PEREIRA JÚNIOR et al., 2020a). Além de muitos utilizarem imagens de baixíssima altitude ($\leq$ 2 m), tiradas a partir do solo (manualmente ou acopladas no maquinário agrícola) ou de alta altitude (imagens de satélites) (HASAN et al., 2021).

Considerando o cenário de grande utilização dos VANTs para obtenção das imagens para a AP e a grande importância da detecção das linhas de plantio para a AP, este projeto tem como foco a detecção de linhas de plantio de plantações de cana-de-açúcar.

Outro ponto importante é que atualmente não foi possível encontrar nenhum *software* de código aberto (ou mesmo gratuito) que faça a detecção automática das linhas de plantio corretamente (em cana-de-açúcar ou outra cultura), apenas *softwares* comerciais, como o InfoRow (2021). Assim, este projeto também busca desenvolver uma metodologia de detecção das linhas de plantio que possa ser utilizada no desenvolvimento de um *software*, promovendo uma democratização do conhecimento.

1.1 Motivação

A segmentação das linhas de plantio apresenta muitos desafios, como o crescimento irregular da cultura, manchas no solo ou solo de variadas tonalidades, falhas no plantio, problemas na cultura (por pisoteamento do maquinário agrícola, efeitos adversos do clima ou incidência de pragas, doenças e ervas daninhas) e condições variadas de iluminação durante a captura das imagens. Além desses desafios, a segmentação no cultivo da cana-de-açúcar tem uma motivação a mais, ser uma cultura semi-perene, onde a cultura ficará por anos produzindo, diferente de outras culturas como milho e beterraba, muito avaliadas em projetos de segmentação das linhas de plantio.

A cana-de-açúcar é utilizada em uma variedade de fins, e, simultaneamente, o Brasil ser o maior produtor da cultura, com milhões de hectares plantados (e recebendo os mais variados e recentes cuidados da AP), motiva diversas pesquisas em como maximizar a produtividade da área plantada. Assim, este projeto tem como principal motivação avaliar os métodos presentes na literatura e propor uma abordagem aberta para identificação das linhas de plantio.

1.2 Objetivos

Este projeto tem como objetivo principal comparar diferentes métodos de segmentação automática em imagens de VANTs com foco em imagens de plantações de cana-de-açúcar e propor melhorias na detecção de linhas de plantio.

Como objetivos específicos deste projeto, tem-se:

- ❏ Estudar e analisar os métodos propostos na literatura correlata para detecção das linhas de plantio, como, por exemplo, Transformada de Hough e Transformada de Radon, Segmentação, Redes Neurais Convolucionais, Índices de Vegetação e Operações Morfológicas.
- ❏ Avaliar a detecção de linhas de plantio utilizando as imagens de VANTs em métodos de *Deep Learning* combinados com Índices de Vegetação.
- ❏ Detectar as linhas de plantio nas imagens das plantações de cana-de-açúcar, com plantas em variados estágios (antes do primeiro corte (cana-planta) e depois do primeiro corte (cana-soca)).
- ❏ Desenvolver um método capaz de identificar, nas imagens obtidas por VANTs, as linhas de plantio com formatos de linhas retas e linhas curvas.
- ❏ Aplicar a proposta em imagens reais de plantações de cana-de-açúcar, analisar e avaliar os resultados obtidos.
- ❏ Desenvolver uma metodologia de detecção das linhas de plantio que possa ser utilizada no desenvolvimento de um *software* gratuito e de código aberto, promovendo uma democratização do conhecimento.

1.3 Hipótese

Este trabalho parte de duas hipóteses principais:

1. Algorítimos ou modelos de *Deep Learning* podem obter melhores resultados na detecção de linhas de plantio do que modelos tradicionais.
2. Índices de Vegetação podem, de alguma maneira, ajudar ou facilitar a detecção de das linhas de plantio.

1.4 Contribuições

As principais contribuições deste trabalho são:

1. Estudo avaliativo de três algorítimos de *Deep Learning* utilizados para segmentação de imagens. Os algorítimos (U-Net, PSPNet e LinkNet) foram aplicados em imagens capturadas por VANTs de plantações variadas de cana-de-açúcar, onde a U-Net obteve os melhores resultados.

2. Avaliação de doze combinações/configurações (i.e., número de blocos e filtros convolucionais) na U-Net de modo a averiguar as melhores combinações para segmentar as imagens de cana-de-açúcar em duas categorias (linhas de plantio e solo/fundo) capturadas por VANTs.

3. Análise de dez Índices de Vegetação como dados de entrada (exclusivos ou complementares) na U-Net, com o objetivo de segmentar as imagens de cana-de-açúcar, resultando em recomendações (e contraindicações) quanto a sua utilização.

4. Disponibilização dos códigos desenvolvidos, que permitem a detecção das linhas de plantio no cenário deste projeto, e que também podem ser utilizados como fonte de consulta ou no desenvolvimento de outros *softwares* com propósitos semelhantes. O código e outros arquivos relacionados podem ser acessados no repositório do GitHub: <https://github.com/ryuuzaki42/Deteccao_de_linhas_de_plantio_em_plantacoes_de_cana-de-acucar_utilizando_deep_learning>.

1.5 Organização da Dissertação

Esta dissertação está estruturada em seis capítulos. A seguir, uma breve descrição de cada capítulo é apresentada:

- ❑ No Capítulo 2 - Revisão da literatura correlata: são apresentados e detalhados os principais conceitos relacionados com o desenvolvimento deste projeto. Dentre eles estão o Processamento Digital de Imagens, Operações Morfológicas, Índices de Vegetação, *Deep Learning* e uma breve descrição do cultivo e características da cana-de-açúcar.

- ❑ No Capítulo 3 - Trabalhos relacionados: são descritos alguns dos principais trabalhos recentes e relacionados com a detecção das linhas de plantio, com foco em plantações de cana-de-açúcar. Esses trabalhos representam o estado da arte e serviram de base para a desenvolvimento deste projeto.

- ❑ No Capítulo 4 - Metodologia de pesquisa: são descritos os *datasets* utilizados neste projeto, as etapas da metodologia do projeto e as métricas utilizadas para a avaliação dos resultados.

- ❑ No Capítulo 5 - Análise e discussão dos resultados: são descritos sete estudos de casos, nos quais a abordagem proposta foi aplicada. Em cada estudo de caso,

os resultados obtidos são discutidos com a finalidade de mensurar a qualidade da abordagem proposta.

- No Capítulo 6 - Conclusão: são descritas as conclusões deste trabalho, limitações, trabalhos futuros e contribuições para a literatura.

Capítulo 2

Revisão da literatura correlata

Neste capítulo são apresentados os principais conceitos e técnicas que fundamentam o presente projeto. Em particular, técnicas de Processamento Digital de Imagens, Índices de Vegetação, *Deep Learning* e informações sobre o cultivo da cana-de-açúcar.

2.1 Processamento Digital de Imagens

O Processamento Digital de Imagens (PDI) compreende a manipulação de imagens digitais por um computador de modo a atingir um objetivo específico com procedimentos envolvendo várias etapas (GONZALEZ; WOODS, 2010).

Após a aquisição das imagens é comum utilizar uma etapa de pré-processamento das imagens, onde são feitos ajustes na intensidade dos pixels (sem conhecimento sobre o que representam). Esses ajustes visam remover imperfeições (e.g., presença de pixels ruidosos, contraste e/ou brilho inadequado) geradas no processo de aquisição e realçar detalhes importantes para análise. Assim, essa etapa tem como objetivo melhorar a imagem original para ser utilizada no processamento principal (MARQUES FILHO; VIEIRA NETO, 1999).

Dependendo do contexto da aplicação, muitas são as técnicas de pré-processamento que podem ser utilizadas, tais como a correção de brilho e contraste, realce de brilho e realce de bordas. A seguir são descritas em detalhes algumas dessas etapas.

2.1.1 Segmentação

A segmentação de imagens tem como objetivo dividir a imagem em regiões ou objetos de interesse. O nível de divisões depende do problema a ser resolvido, sendo a segmentação executada até que todos os objetos de interesse sejam detectados. A segmentação de imagens é considerada uma das tarefas mais difíceis do PDI, já que implica diretamente no resultado da etapa de análise. Assim, é importante selecionar qual a técnica de segmen-

tação mais adequada para o problema em análise, de modo a aumentar a probabilidade de se obter uma segmentação precisa (GONZALEZ; WOODS, 2010).

Ao fim da etapa de segmentação, a imagem é dividida em várias regiões e sem sobreposição entre elas, ou seja, uma parte da imagem pertence somente a uma região de interesse. O tipo de segmentação utilizada depende do problema a ser tratado, por exemplo, se for necessário dividir a imagem em regiões sem rotulá-las (segmentação clássica), ou se for necessário rotular cada pixel da imagem a uma determinada classe (segmentação semântica). Dentre as técnicas de segmentação, a limiarização é uma das mais utilizadas, por ser eficiente do ponto de vista computacional (KURUVILLA et al., 2016).

A limiarização (*Thresholding* em inglês) é uma forma bastante simples de segmentação, que consiste na divisão da imagem em duas classes (fundo e objeto) a partir de um limiar T. Caso o valor do pixel seja maior ou igual a T, terá valor 1 na imagem gerada; caso contrário, receberá o valor 0. Como resultado, esse processo produzirá uma imagem binária (com pixels de valor 0 ou 1), processo também chamado de binarização (MARQUES FILHO; VIEIRA NETO, 1999).

A escolha do limiar é uma tarefa complicada, já que um limiar inadequado não conseguirá separar de forma correta a imagem em duas classes. Nesse contexto, a aplicação de limiares locais (i.e., um para cada pequeno retângulo da imagem de modo a compensar a não uniformidade de iluminação e/ou da refletância), pode ser uma opção melhor. Outra abordagem é a utilização de múltiplos limiares globais, onde a imagem é divida em função dos intervalos dos limiares em mais de duas classes (GONZALEZ; WOODS, 2010).

O limiar pode ser escolhido manualmente ou através do uso de algum algoritmo. Para a escolha desse limiar, muitos trabalhos na área de imagens utilizam o método de Otsu. Esse método (OTSU, 1979) utiliza o histograma da imagem e busca um limiar ideal que separe a imagem em duas classes (objeto e fundo), de modo a maximizar a variância entre as classes e minimizar a variância interna das classes.

2.1.2 Operações Morfológicas

O termo morfologia é comumente utilizado no ramo da biologia para se referir ao estudo das formas e estruturas dos animais e das plantas. No contexto de processamento de imagens, a morfologia matemática tem sentido similar, sendo utilizada para extrair componentes das imagens que são úteis na representação e na descrição da forma de uma região. A morfologia matemática oferece operações úteis para resolver vários problemas do PDI, como a detecção de bordas, a segmentação e o realces de objetos presentes na imagem (MARQUES FILHO; VIEIRA NETO, 1999).

A morfologia matemática utiliza a teoria dos conjuntos, onde os conjuntos representam objetos presentes na imagem. Em imagens binárias, os conjuntos são membros do espaço inteiro bidimensional Z^2 (onde cada elemento é um vetor de coordenadas (x, y)), já em

imagem em escala de cinza, os elementos são membros do Z^3, sendo os dois primeiros as coordenadas do pixel e o terceiro, seu nível de cinza (GONZALEZ; WOODS, 2010).

As operações morfológicas utilizam os chamados elementos estruturantes (pequenos conjuntos ou subimagens) e as operações de conjunto (e.g., a reflexão e a translação) como base. A translação é definida pela Equação 1, onde os pontos em B, com as coordenadas (x, y) foram substituídas por $(-x, -y)$. Por outro lado, a reflexão é definida pela Equação 2, onde os pontos em B, com as coordenadas (x, y) foram substituídas por $(x + z_1, y + z_2)$ (ALAZAWEE; ABDEL-QADER; ABDEL-QADER, 2015).

$$\hat{B} = \{w | w = -b, \text{ para } b \in B\} \tag{1}$$

$$(B)_z = \{c | c = b + z, \text{ para } b \in B\} \tag{2}$$

Dentre as operações morfológicas, as quatro mais utilizadas são: a dilatação, a erosão, a abertura e o fechamento. As duas primeiras são mais básicas, enquanto as duas últimas são combinações das primeiras. Considere A e B como conjuntos de Z^2 e assim sendo imagens binárias. A dilatação, definida pela Equação 3, é responsável pelo aumento ou crescimento dos limites dos objetos na imagem em função do elemento estruturante. Dessa forma, a dilatação pode ser utilizada para preencher lacunas entre os objetos de interesse. Por outro lado, a erosão, definida pela Equação 4, diminui a quantidade de pixels de interesse em torno dos objetos, dando origem às lacunas. Desse modo, a erosão pode ser utilizada para separar objetos que estejam ligados por pequenos filamentos (MARQUES FILHO; VIEIRA NETO, 1999).

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \varnothing\} \tag{3}$$

$$A \ominus B = \{z | (B)_z \subseteq A\} \tag{4}$$

A aplicação de uma dilatação seguida por uma erosão (com o mesmo elemento estruturante) é definida como fechamento morfológico. Em contrapartida, a abertura é o oposto, uma erosão seguida de uma dilatação. A abertura e o fechamento geralmente suavizam os contornos dos objetos presentes na imagem, mas a abertura tende a quebrar trechos muito obtusos e estreitos, enquanto o fechamento tende a fechar pequenas lacunas e preencher falhas de um contorno (GONZALEZ; WOODS, 2010; KAUR; SAHAMBI, 2015).

2.1.3 Índices de Vegetação

Os Índices de Vegetação (IVs) são combinações algébricas de várias bandas espectrais, de modo a destacar a vegetação e suas propriedades (e.g., quantidade de biomassa, ausência de determinados nutrientes, deficiências hídricas, porcentagem de cobertura do solo)

(CANDIAGO et al., 2015). Cada IV tem foco em certas características da vegetação e um momento mais oportuno para seu uso. Dentre eles, o índice de excesso de verde (do inglês *Excess Green index* (ExG)) é um dos mais utilizados para câmeras *Red*, *Blue* e *Green* (RGB) (PEREIRA JÚNIOR et al., 2020a). O RGB corresponde as bandas RGB de uma imagem, R (*Red* - vermelho), G (*Green* - verde) e B (*Blue* - azul).

Para o cálculo do índice ExG, primeiramente cada banda da imagem RGB é normalizada no intervalo de $[0, 1]$, pela Equação 5, onde R_{max}, G_{max} e B_{max} são os valores máximos correspondentes de cada banda, que geralmente é 255 para imagens de 24 bits de cor. Na sequência, são calculadas as coordenadas cromáticas, r, g e b pela Equação 6. Por fim, o ExG é calculado pela Equação 7 (GARCÍA-SANTILLÁN et al., 2017). Dentre os muitos IVs desenvolvidos para imagens RGB, na Tabela 1 pode ser visualizado alguns dos mais utilizados (LU et al., 2019).

$$R_n = \frac{R}{R_{max}}, G_n = \frac{G}{G_{max}}, B_n = \frac{B}{B_{max}} \tag{5}$$

$$r = \frac{R_n}{R_n + G_n + B_n}, g = \frac{G_n}{R_n + G_n + B_n}, b = \frac{B_n}{R_n + G_n + B_n}, \text{ onde } r + g + b = 1 \tag{6}$$

$$\text{ExG} = 2 * g - r - b \tag{7}$$

Para câmeras com espectro infravermelho próximo (do inglês *Near Infrared* (NIR)), o índice de vegetação de diferença normalizada (do inglês *Normalized Difference Vegetation Index* (NDVI)) é um dos mais utilizados (PEREIRA JÚNIOR et al., 2020a). O NDVI é calculado pela Equação 8, onde NIR é a banda do infravermelho próximo (CANDIAGO et al., 2015).

$$\text{NDVI} = \frac{NIR - R}{NIR + R} \tag{8}$$

2.2 *Deep Learning*

O aprendizado profundo (do inglês *Deep Learning* (DL)) é uma subárea do Aprendizado de Máquina, com aprendizado por representações a partir dos dados e aprendizado continuado através das várias camadas (*layers*). Essas camadas dão origem ao termo *deep* em *Deep Learning*, caracterizando assim redes profundas, onde elas apreendem abstrações cada vez mais significativas sobre os dados. Essas redes neurais artificiais se tornaram o estado da arte para vários problemas de Visão Computacional e PDI. Dentre elas, as redes neurais convolucionais (do inglês *Convolutional Neural Networks* (CNNs)) são uma das mais conhecidas e utilizadas (PONTI et al., 2017).

As CNNs podem utilizar vários tipos de camadas na sua construção (Figura 1), sendo que as principais são (RAWAT; WANG, 2017):

Tabela 1 – Índices de Vegetação (LU et al., 2019)

IV	Nome	Equação
VARI	*Visible Atmospherically Resistant Index*	$\text{VARI} = \frac{g - r}{g + r - b}$
ExG	*Excess Green Index*	$\text{ExG} = 2 * g - r - b$
ExR	*Excess Red Vegetation Index*	$\text{ExR} = \frac{1,4 * R - G}{G + R + B}$
ExB	*Excess Blue Vegetation Index*	$\text{ExB} = \frac{1,4 * B - G}{G + R + B}$
ExGR	*Excess Green minus Excess Red*	$\text{ExGR} = \text{ExG} - \text{ExR}$
GRVI	*Green Red Vegetation Index*	$\text{GRVI} = \frac{G - R}{G + R}$
MGRVI	*Modified Green Red Vegetation Index*	$\text{MGRVI} = \frac{G^2 - R^2}{G^2 + R^2}$
GLI	*Green Leaf Index*	$\text{GLI} = \frac{2 * g - r - b}{-r - b}$
RGBVI	*Red Green Blue Vegetation Index*	$\text{RGBVI} = \frac{G^2 - B * R}{G^2 + B * R}$
IKAW	*Kawashima Index*	$\text{IKAW} = \frac{R - B}{R + B}$

R, G e B correspondem as bandas RGB de uma imagem.
r, g e b são as coordenadas cromáticas, pela Equação 6.

- **camada convolucional**: composta por um conjunto de filtros, onde cada filtro é uma matriz com pesos apreendidos pela rede a partir das informações de entrada (e.g., uma imagem e seu respectivo rótulo). Inicialmente, os pesos são, geralmente, definidos aleatoriamente, sendo atualizados pela interação da rede com os dados de entrada e o erro observado. A partir de cada filtro aplica-se uma convolução (Figura 2) (dando nome a rede) nas imagens de entrada. Desse modo, os filtros funcionam como extratores de características e ficam cada vez mais especializados em certas características (e.g., bordas, linhas) presentes nas imagens. A camada convolucional geralmente é seguida por uma função de ativação, sendo a função ReLu (do inglês *Rectified Linear Unit*) uma das mais utilizadas pelos seus bons resultados nos treinamentos das redes. A função retorna 0 se receber uma entrada negativa, mas para qualquer outro valor ela retorna esse mesmo valor.

- **camada de *pooling***: recebe o resultado das camadas anteriores, reduzindo seu tamanho através de uma função de *pooling*. Uma função de *pooling* substitui a saída da rede em um determinado local por uma estatística resumida das saídas próximas. Por exemplo, a operação de *max pooling* (Figura 4) retorna como saída o valor máximo na vizinhança (geralmente 2×2) analisada. Assim, essa camada tem por objetivo reduzir o custo computacional da rede, facilitando o seu treinamento

e sua utilização. Além de possibilitar a invariância espacial, onde a rede conseguirá trabalhar com distorções e translações nas imagens de entrada.

- **camada totalmente conectada**: esse tipo de camada geralmente é utilizada após várias camadas de convolução e de *pooling*. Nela tem-se a ligação do resultado da camada anterior e seus pesos para, por exemplo, classificar a imagem de entrada em determinada categoria. A partir do erro observado, os pesos das camadas anteriores são atualizados.

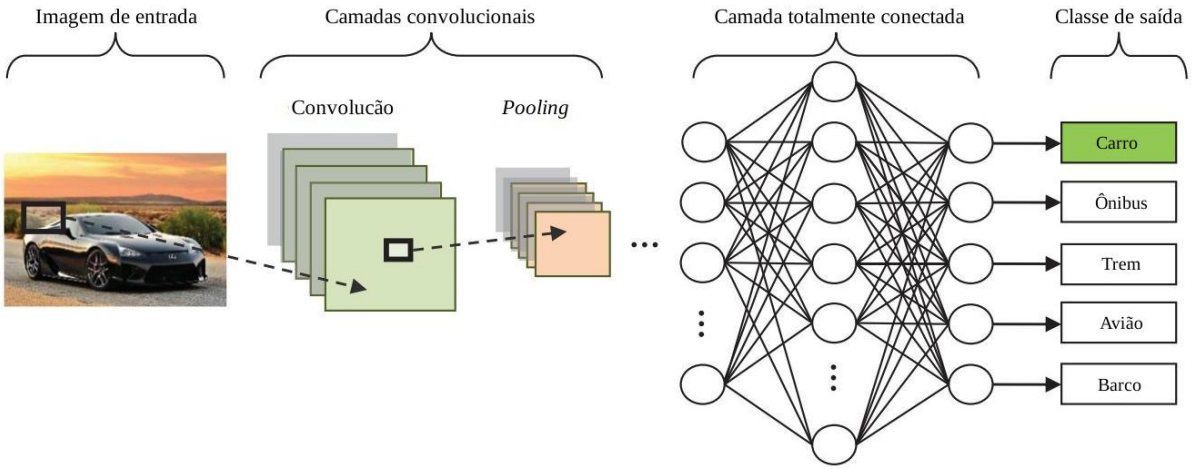


Figura 1 – Modelo visual da arquitetura de uma CNN . Adaptado de Rawat e Wang (2017).

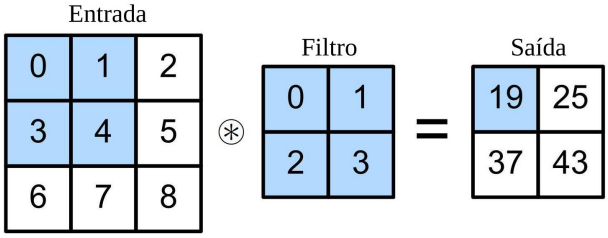


Figura 2 – Exemplo de convolução ($\circledast$) em uma entrada 2D de $3 \times 3 \times 1$ com um filtro $2 \times 2 \times 1$ já rotacionado em 180^{o}, com *stride* de 1 (para altura e largura) e sem *padding*. Operação para a posição $(0, 0) = 0 * 0 + 1 * 1 + 3 * 2 + 4 * 3 = 19$. Adaptado de Zhang et al. (2021).

Algumas das arquiteturas de CNNs mais conhecidas incluem AlexNet, VGGNet e ResNet (MINAEE et al., 2021). Uma CNN consegue classificar uma imagem para uma das várias categorias que foi treinada ou reconhecer a presença de um objeto em qualquer parte da imagem analisada. Por outro lado, para a segmentação de imagens a informação espacial precisa ser preservada, por isso outros modelos de redes (e suas variações) são utilizados (PONTI et al., 2017).

Muitos modelos de DL foram, e estão sendo, desenvolvidos para segmentação de imagens, e.g., redes totalmente convolucionais, U-Net, LinkNet e PSPNet. Nelas, a saída da rede tem o mesmo tamanho da entrada (ou um pouco menor), e representa o resultado da segmentação da imagem de entrada. Elas podem ser treinadas a partir do zero com os dados a serem segmentados, ou utilizar modelos pré-treinados (*transfer learning*) com outros *datasets* (como o ImageNet), além de poderem ser utilizados variados *backbones* como extratores de características (HAO; ZHOU; GUO, 2020; MINAEE et al., 2021). ImageNet é um *dataset* com mais de 14 milhões de imagens pertencentes a mais de 22 mil categorias, com uma média de mil imagens em cada categoria (RUSSAKOVSKY et al., 2015; ZHENG et al., 2019).

Dentre as CNNs, a VGGNet e a ResNet são as mais utilizadas como *backbones* (LATEEF; RUICHEK, 2019). A VGG16 (SIMONYAN; ZISSERMAN, 2015), proposta pelo grupo VGG (*Visual Geometry Group*) da Universidade de Oxford, tem bom desempenho de generalização e de extração de características. Deste modo, é capaz de obter uma melhor precisão na classificação ao utilizar como base um modelo pré-treinado com um *dataset*, como, por exemplo, o ImageNet. Na VGG16 geralmente se utiliza *kernels* de convolução 3×3 e *stride* de 1 pixel, de modo a otimizar o desempenho da rede, além de ser muito utilizada como *backbone*, por exemplo, para aplicações de AP (FAWAKHERJI et al., 2019).

As redes totalmente convolucionais (do inglês *Fully Convolutional Netwoks* (FCNs)) (Figura 3) são baseadas nas CNNs, onde a camada totalmente conectada é removida e são adicionadas camadas de *unpooling* e de *transposed convolution* (convolução transposta). O *unpooling* (Figura 4), ao contrário do *pooling*, gera um mapa de características de maior resolução como saída. Outrossim, a *transposed convolution* (Figura 5), às vezes chamada de *deconvolution*, mesmo que a operação executada não seja o oposto da convolução, também é aplicada de modo a aumentar a resolução do mapa de características (LONG; SHELHAMER; DARRELL, 2015; FANG, 2017).

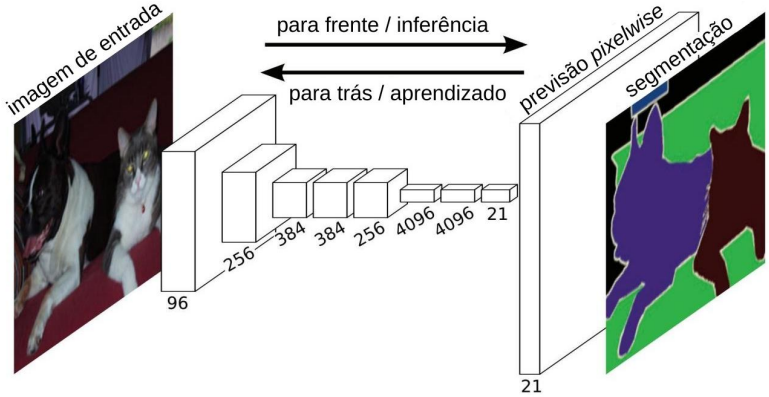


Figura 3 – Arquitetura simplificada de uma FCN. Adaptado de Long, Shelhamer e Darrell (2015).

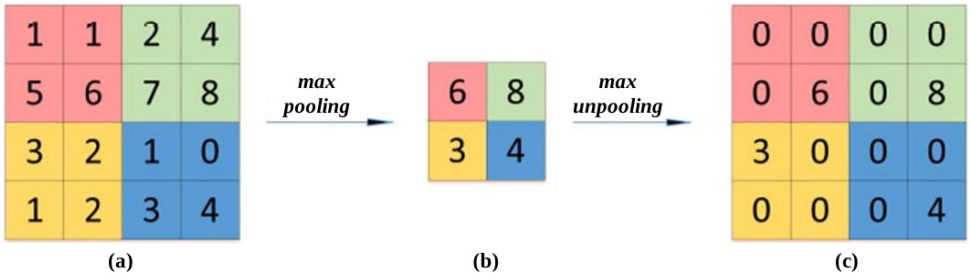


Figura 4 – Exemplo de *pooling* e *unpooling*: (a) entrada 2D de $4 \times 4 \times 1$; (b) resultado da aplicação do *max pooling* com janela de 2×2 e *stride* de 2; (c) resultado da aplicação do *max unpooling*, a posição dos valores é guardada na operação de *pooling*. Adaptado de Fang (2017).

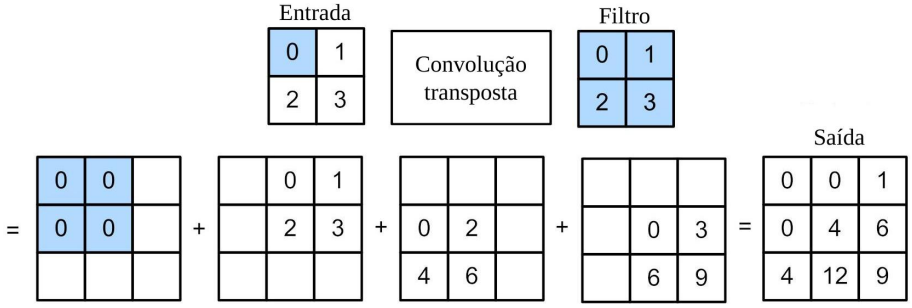


Figura 5 – Exemplo de convolução transposta em uma entrada 2D de $2 \times 2 \times 1$ com um filtro $2 \times 2 \times 1$, com *stride* de 1 (para altura e largura) e sem *padding*. Adaptado de Zhang et al. (2021).

A U-Net (Figura 6), usada inicialmente para segmentar imagens médicas, foi baseada nas FCNs. Ela apresenta um formato em "U" (dando nome a rede), sendo simétrica: na primeira metade tem-se os *encoders*, reduzindo os mapas de características e na outra metade os *decoders*, expandindo eles até o tamanho de entrada original. A primeira parte é a *contraction path* e a segunda parte é a *expansive path*, devido às operações realizada em cada parte, reduzindo ou expandindo o tamanho da imagem processada. A U-Net tem *skip connection* (linhas em cinza na Figura 6) como as FCNs, contudo aqui os mapas de características também são concatenados com seu simétrico na outra metade (RONNEBERGER; FISCHER; BROX, 2015; HAO; ZHOU; GUO, 2020)

A LinkNet (Figura 7) é similar a U-Net, porém utiliza *residual blocks* (blocos residuais) nos *encoders* e nos *decoders*. Ela foi desenvolvida tendo como foco a segmentação semântica, com a arquitetura da ResNet sendo usada como base. Os *residual blocks* (Figura 8), também chamados de *shortcut connections*, são úteis no problema de *vanishing gradient* (dissipação do gradiente) em redes muito profundas, pois enviam o resultado para uma ou mais camadas a frente (CHAURASIA; CULURCIELLO, 2017; EBRAHIMI; ABADI, 2021).

A *Pyramid Scene Parsing Network* (PSPNet) (Figura 9) utiliza o *pyramid pooling module* (módulo de *pooling* em pirâmide), que explora informações de contexto global através

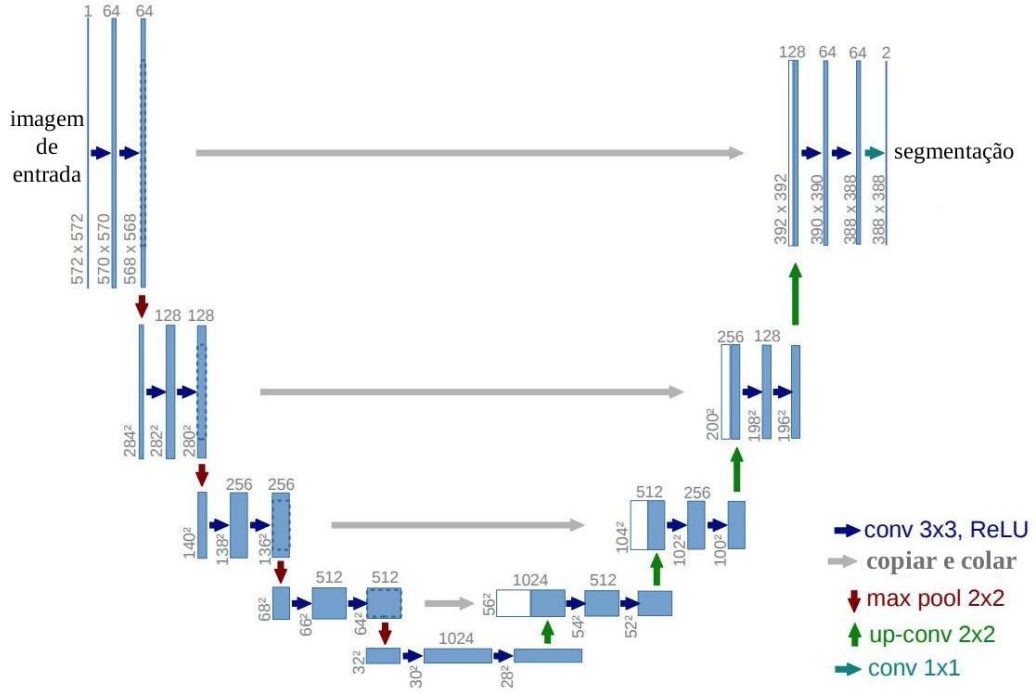


Figura 6 – Arquitetura da U-Net. Adaptado de Ronneberger, Fischer e Brox (2015).

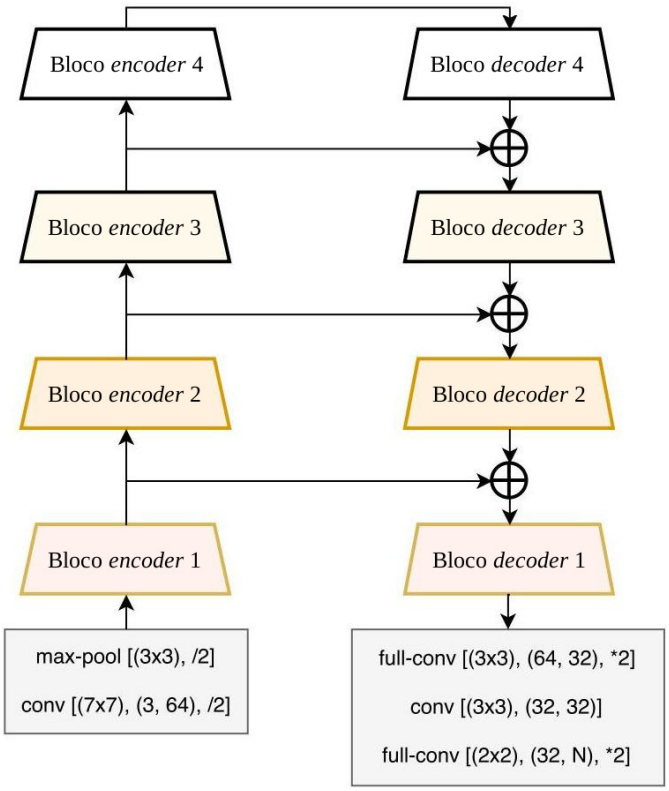


Figura 7 – Arquitetura da LinkNet. Adaptado de Chaurasia e Culurciello (2017).

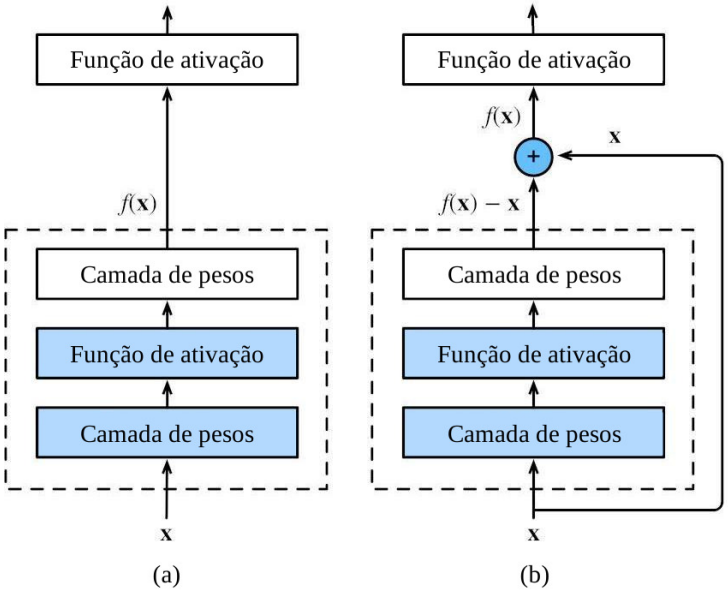


Figura 8 – Exemplo de bloco regular e bloco residual: (a) bloco regular, com os dados passando pelas camadas (i.e., camada de pesos (*weight layer*) e função de ativação (*activation function*)); (b) bloco residual, onde a linha sólida leva a entrada $\mathbf{x}$ até operador de adição, propagando mais rápido os dados pela rede. Adaptado de Zhang et al. (2021).

de diferentes regiões baseadas em agregação de contexto. O módulo funciona aplicando várias funções de *pooling* de tamanhos diferentes na saída da camada de convolução da rede, produzindo diversas representações hierárquicas da imagem. As informações locais e globais são agrupadas para fazer uma predição final de maior confiança. Por base, a PSPNet utiliza uma rede ResNet pré-treinada utilizando *dilated convolution* (convoluções dilatadas) como extratores de características.

A *dilated convolution* (Figura 10) é um tipo de convolução que dilata o filtro ao inserir espaços entre os valores do mesmo. Os espaços são definidos pelo parâmetro r, a taxa de dilatação, sendo que $r = 1$ é a convolução tradicional, já para $r > 1$ são convoluções dilatadas. Elas podem ser entendidas pela inserção de $r - 1$ zeros entre cada dois pesos adjacentes em um filtro convolucional tradicional (ZHAO et al., 2017; WANG; JI, 2021).

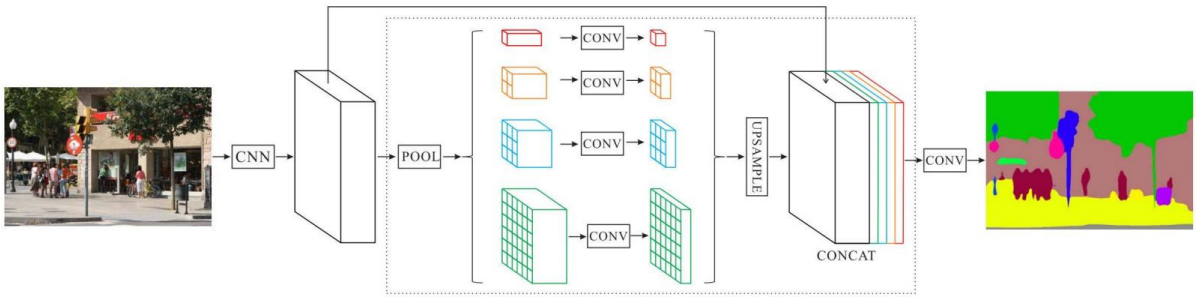


Figura 9 – Arquitetura da PSPNet. Adaptado de Zhao et al. (2017).

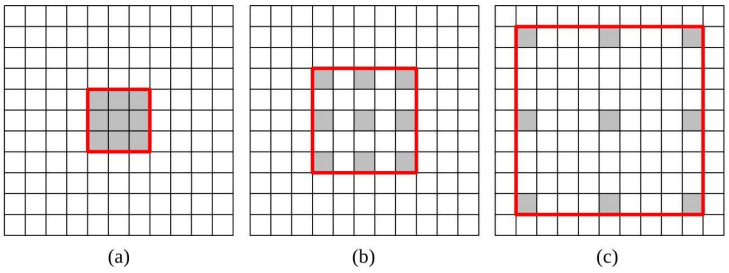


Figura 10 – Exemplos de convolução dilatada com um filtro 3×3. Os pontos em cinza são utilizados na operação: (a) $r = 1$, convolução tradicional; (b) $r = 2$, dilatada por 1 espaço; (c) $r = 4$, dilatada por 3 espaços. Adaptado de Wang e Ji (2021).

2.3 Cana-de-açúcar

A cana-de-açúcar (*Saccharum spp.*) é uma cultura semi-perene, o que significa que a cultura pode ser colhida por anos até que seja necessário um novo plantio. A parte colhida são os colmos, ou também chamados de caules. A primeira produção após o plantio é chamada de cana-planta. As demais produções são chamadas de cana-soca, ou seja, colmos provenientes da rebrota (nova brotação) da soqueira (Figura 11). A colheita da rebrota das soqueiras pode ser feita por 4 ou mais anos (SEGATO et al., 2006; RUDORFF et al., 2010).

O plantio da cana-de-açúcar geralmente é feito com toletes, que são colmos fracionados com cerca 3 ou 4 gemas cada, em seguimentos de linha seguindo o relevo do terreno. O espaçamento entre as linhas de plantio pode variar de 0,9 a 1,6 m, dependendo do terreno, assim é esperado encontrar fileiras da cultura e variadas partes de solo exposto. A produção da cultura usualmente tem uma redução gradual a cada colheita até um nível antieconômico, deste modo requer um novo plantio. Essa redução pode ter várias causas, como: fatores climáticos; ocorrência de pragas; pisoteio por máquinas e veículos que prejudicam a brotação e o crescimento da planta; e falhas na cultura (SEGATO et al., 2006; MAY; RAMOS, 2019).

As culturas de cana-de-açúcar podem ser classificadas em 4 fases de desenvolvimento (brotação e estabelecimento, perfilhamento, crescimento de colmos e maturação), como pode ser visualizado na Figura 12. O perfilhamento é o processo de emissão de colmos por uma mesma planta, denominados perfilhos. Na maturação os colmos amadurecem até a época da colheita. Após o corte, da cana-planta ou cana-soca, inicia-se um novo ciclo de aproximadamente 12 meses (MOLIJN et al., 2019).

Com os avanços na AP com a utilização VANTs, imagens de plantações estão sendo utilizadas para detecção das linhas de plantio, mapeamento de falhas, estimativa de produção, detecção de pragas, entre outros (DOHA et al., 2021). Dentre as fases de desenvolvimento e seus respectivos dias, Souza et al. (2017) recomendam que as imagens

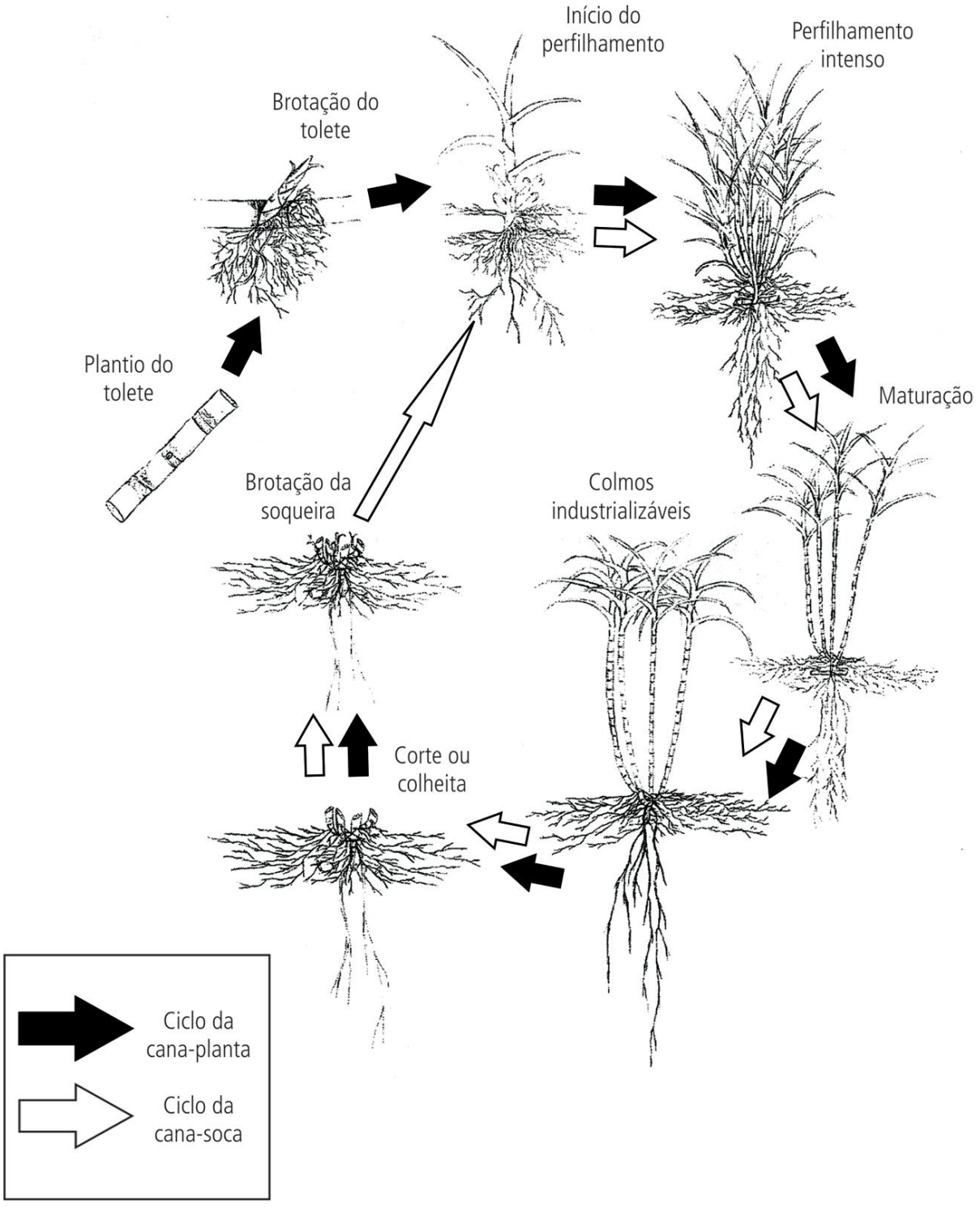


Figura 11 – Ciclos de desenvolvimento da cana-de-açúcar (SEGATO et al., 2006).

das plantações de cana-de-açúcar sejam capturadas entre 60 e 90 dias após o plantio ou corte (colheita), quando a cana-de-açúcar já cresceu o suficiente para preencher a linha de plantio, mas sem cobrir os espaços entre as linhas. Entretanto, esse período pode variar dependendo da variedade de cana-de-açúcar plantada, da quantidade de plantas/socas na linha, da quantidade de chuva, dentre outros fatores. Assim, também é recomendado

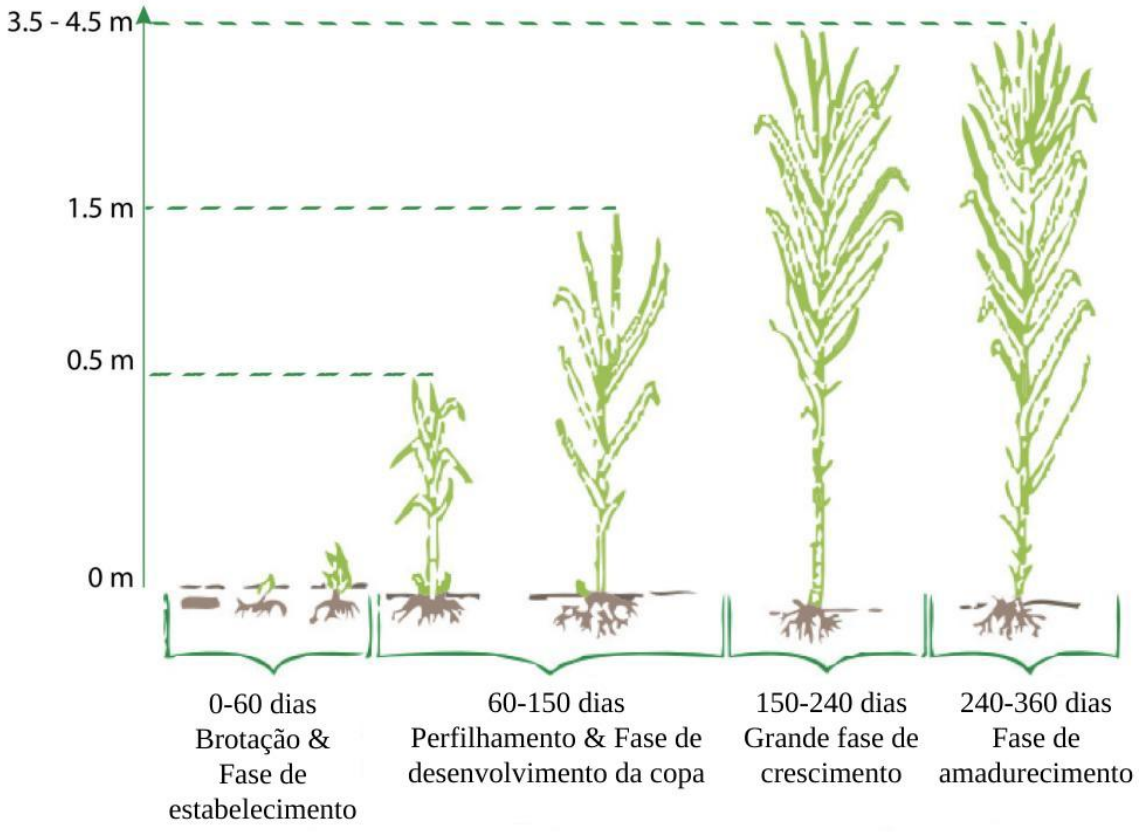


Figura 12 – Fases de desenvolvimento da cana-de-açúcar (Tamanho × Dias). Adaptado de Molijn et al. (2019).

uma análise visual da plantação para confirmação do melhor momento para capturar as imagens.

Em relação à detecção de falhas nas linhas de plantio, Oliveira et al. (2023) têm recomendações bem similares, que a altura da planta pode afetar significativamente o mapeamento utilizando os VANTs. Para evitar superestimação dos resultados (falhas detectadas) quando as plantas são pequenas ou subestimação quando as plantas são altas, é necessário realizar o mapeamento no momento do desenvolvimento da cultura.

Em relação ao horário para capturar as imagens, Barbosa Júnior et al. (2022) recomendam realizar os voos às 12:00 (meio-dia), por se mostrar o horário mais confiável com base nas variáveis condições de iluminação ao longo do dia. Os voos devem durar até 2 horas e serem realizados em altas altitudes para conseguir capturar imagens de uma maior área em menor tempo. Entretanto, é necessário um equilíbrio entre qualidade das imagens requisitadas e a disponibilidade para capturar as imagens. Também recomendam dividir a plantação em partes semelhantes e realizar as análises e tomadas de decisão restrita a elas, não para a plantação como um todo.

Considerando que as imagens foram capturadas nos períodos recomendados, é esperado ter uma boa visibilidade das linhas de plantio e das partes de solo exposto. Desse modo, dois cenários são mais esperados: com cana-planta e cana-soca, como na Figura 13. No

estágio de cana-soca, o solo está coberto com folhas secas do último corte/colheita, dando aspecto amarelado, dificultando a análise computacional, já no estágio de cana-planta o solo está mais visível.

(a) (b)

Figura 13 – Exemplos de plantações de cana-de-açúcar: (a) estágio de cana-planta (após o plantio e o nascimento, mas antes do primeiro corte); (b) estágio de cana-soca (após o corte) (SILVA, 2020).

Capítulo 3

Trabalhos relacionados

Neste capítulo são discutidos alguns dos principais trabalhos recentes e relacionados com este projeto. Tais trabalhos serviram de base para o desenvolvimento de novas metodologias para a segmentação de imagens de VANTs associadas à AP e à detecção de linhas de plantio.

Souza et al. (2017) desenvolveram um método para detectar linhas de plantio em imagens de VANTs de plantações de cana-de-açúcar (GSD de aproximadamente 0,10 m) e suas falhas. Para a análise das imagens e para a execução do processo de Análise de Imagem Baseada em Objeto, do inglês *Object-Based Image Analysis* (OBIA), foi utilizado o *software* comercial *eCognition Developer*[1] (*Trimble GeoSpatial*). A OBIA primeiramente identifica unidades espacialmente e espectralmente homogêneas, chamadas de "objetos", criados pelo agrupamento de pixels adjacentes e, em seguida, os utiliza como elementos básicos para análise. Para identificar as linhas de plantio foi utilizado o NDVI (as imagens utilizadas têm as bandas RGB e a NIR). Também foi utilizado o *software* comercial *Arcgis*[2] (*Esri*) para extrair automaticamente falhas nas linhas (através do "*ModelBuilder*"). O método foi avaliado comparando 54 falhas estimadas pelo método e as observadas (medidas por fita métrica). Apesar dos resultados estimados serem próximos dos observados, o método apresenta limitações em relação aos *softwares* comerciais utilizados e poucos testes que foram realizados.

O trabalho de García-Santillán et al. (2017) utiliza imagens de baixíssima altitude (2 m, câmera acoplada no maquinário agrícola) para identificar linhas em plantações de milho. A partir das imagens capturadas, apenas uma região (de interesse) é utilizada. Para a segmentação foi utilizado o índice de vegetação ExG (para facilitar a diferenciação de plantas e ervas daninhas do solo). Na sequência é aplicado o método de Otsu com dois limiares (dividindo a imagem em 3 classes: milho, erva daninha e solo) e depois operações morfológicas são aplicadas. Após a segmentação, a Transformada de Hough (TH) foi

[1] Disponível em: <https://geospatial.trimble.com/products-and-solutions/ecognition>. Acesso em: 7 de março de 2023

[2] Disponível em: <https://www.esri.com/en-us/arcgis/products/arcgis-pro/overview>. Acesso em: 7 de março de 2023

utilizada para identificar 4 pontos, os quais são o começo de 4 linhas de plantio presentes na imagem. A partir deles faz-se uma projeção de 4 linhas retas até o meio da imagem (dividindo em sub-regiões de interesse). Em seguida, um algoritmo é utilizado para ajustar a inclinação das linhas em microrregiões e expandir para a parte superior da imagem. O método foi testado com marcações de um especialista e teve bom resultado, contudo ele é limitado ao cenário que foi proposto (imagens de baixas altitudes, com 4 linhas de plantio na parte inferior da imagem e linhas ligeiramente curvas na parte superior).

Soares, Abdala e Escarpinati (2018) comentam que a TH é uma boa escolha inicial para detectar linhas de plantio, justificando vários trabalhos anteriores utilizarem esse método. Contudo, a TH necessita que o formato do objeto a ser detectado seja conhecido antecipadamente, o que limita sua utilização para detectar as linhas de plantio. Para mitigar essa limitação, no trabalho utilizam a TH com o esquema janelamento (*tiling scheme*), onde as linhas de plantio se aproximam de linhas retas. No método desenvolvido a TH foi aplicada com o janelamento com variados tamanhos de janela e porcentagem de sobreposição. O método foi avaliado em 8 imagens capturadas por VANTs em plantações de café e suas marcações feitas por um especialista. Os melhores resultados foram com janela de 200 pixels e sobreposição de 48%. A sobreposição ajuda evitar descontinuidades nas linhas detectadas.

No trabalho de Rozo (2019) foi desenvolvido um *plugin* para o *software* de código aberto QGIS[3] (*QGIS Development Team*) para identificar linhas de plantio. O trabalho teve como foco imagens de plantações de cana-de-açúcar capturadas por VANTs. O *plugin* utiliza uma "semente" inserida pelo usuário, que indica a direção das linhas de plantio no mosaico. Posteriormente o mosaico é dividido em pequenos retângulos e em seguida processados. Em cada um deles é aplicado em sequência o ExG, o método de Otsu e as operações morfológicas. Depois a imagem (pequeno retângulo) é binarizada e contornos retangulares são adicionados nos segmentos que representam as plantas. A partir desses segmentos é calculado a orientação da linha de plantio dentro deles e depois expandido e processado (utilizando a "semente") para toda a imagem. O método/*plugin* consegue identificar boa parte das linhas de plantio nos mosaicos (com GSD aproximado de 2 cm) testados, contudo, ele precisa que o mosaico tenha as linhas de plantio em uma mesma direção, o que difere de cenários reais.

Bah, Hafiane e Canals (2020) propuseram uma abordagem utilizando *deep learning*, que chamaram de *CRowNet*. Nela foi utilizada uma FCN, a SegNet, que tem a arquitetura composta por *encoders* (VGG16) e *decoders*. A SegNet é aplicada para segmentar as imagens (RGB) e tem como resultado uma imagem binária (para cada uma delas), indicando os locais mais prováveis das linhas de plantio. Na sequência, uma combinação da TH com uma CNN (5 convoluções dilatadas com 32 filtros cada) foi utilizada, de nome *HoughCNet*. Várias linhas são detectadas pela TH e depois a CNN foi utilizada para

[3] Disponível em: <https://qgis.org/>. Acesso em: 13 de março de 2023

detectar as principais linhas (as mais longas), que os autores consideram como sendo as linhas de plantio. A *CRowNet* foi testada em imagens de VANTs de plantações de beterraba (GSD aproximado de 1 cm) e milho (sem informação do GSD), tendo bom resultado na maioria delas.

Em Silva (2020) foram desenvolvidas duas abordagens para detecção de linhas de plantio de cana-de-açúcar, a primeira utilizando Algoritmo Genético (AG) resultando em Silva, Escarpinati e Backes (2021) e a segunda utilizando redes neurais artificiais. Para comparar os resultados, foram utilizados 4 *datasets* (com plantas de variadas idades) e suas respectivas linhas de plantio, marcadas por um especialista. O AG foi utilizado para encontrar uma máscara (filtro convolucional que combina as bandas RGB da imagem para uma em escala de cinza), que maximize o Coeficiente de *Dice* (CD). Em sequência, o método Otsu foi utilizado para binarizar as imagens e por fim, a Transformada de Radon (TR) foi utilizada para refinar a detecção das linhas de plantio. Foram obtidos valores de CD de 0,56 até próximo de 0,77 em função do *dataset* utilizado na análise e da configuração utilizada no método de Otsu. Como resultado, a detecção das linhas ficou próxima da marcação do especialista, lidando melhor com linhas retas e sendo o seu desempenho inferior em linhas curvas (gerando um efeito de serrilhamento indesejado). Na abordagem com redes neurais, a LinkNet apresentou melhor resultado que na abordagem anterior, com 0,90 de CD no *dataset* utilizado no treinamento. A TR também foi aplicada nessa abordagem, contudo piorando o desempenho em muitos casos.

Oliveira (2020) desenvolveu uma abordagem similar a Silva (2020) para a detecção das linhas de plantio de cana-de-açúcar. Nela, o mosaico é dividido em pequenas imagens, que são convertidas para escala de cinza e realizada a redução dos níveis de cinza (de 256 para 32) pela Transformada Discreta de Wavelet. Na sequência, o AG é utilizado para encontrar 4 limiares, o maior é usado para segmentar a imagem, que segue para operações morfológicas de fechamento, erosão e esqueletização. Por fim, a Transformada de Hough Probabilística Progressiva é utilizada para otimizar a detecção das linhas de plantio e suas falhas. O trabalho apresenta um bom resultado, mas foi aplicado apenas em uma pequena seção do mosaico, sem um padrão ouro para teste e considera as linhas de plantio como linhas retas.

Pereira Júnior et al. (2020a) realizaram uma revisão sistemática, de 2008 até 2018, com foco nos métodos de segmentação de linhas de plantio em imagens de plantações capturadas por meio de VANTs, bem como nos métodos extratores de características utilizados. Dentre os métodos tradicionais, *Support Vector Machine* (SVM), *Random Forest* (RF), *Mahalanobis classifier* (MC) e *K-nearest neighbors* (KNN), foram encontrados como mais utilizados e em trabalhos mais recentemente as CNNs. Para os extratores de características, os IVs e *Gray Level Co-occurrence Matrices* (GLCM) são os mais utilizados, sendo incluídos na comparação, em conjunto com filtros de Gabor, nos métodos tradicionais de segmentação. Os experimentos foram realizados utilizando o mosaico Pereira Júnior e

Wangenheim (2019) de plantações de canas-de-açúcar e suas respectivas marcações realizadas por um especialista. O melhor resultado foi obtido utilizando Linear SVM com RGB + ExG + Gabor, com 0,88 de *F1-score*. Corroborando a ideia de que IVs podem ser úteis para diferenciar as plantas do solo ao conseguir extrair características únicas deles.

Em Pereira Júnior et al. (2020b) realizaram uma comparação dos métodos tradicionais de segmentação e as CNNs para segmentar ervas daninhas em imagens de plantações capturadas por VANTs. Os experimentos foram realizados utilizando dois mosaicos (Pereira Júnior e Wangenheim (2019) e Monteiro e Wangenheim (2019)) de plantações de canas-de-açúcar e suas respectivas marcações realizadas por um especialista. As imagens são segmentadas em três classes: planta/cultura, ervas daninhas e solo. Em relação as CNNs, foram selecionadas quatro redes (SegNet, U-Net, *Full-Resolution Residual Network* (FRRN) e PSPNet) para comparação com os métodos tradicionais. Dentre os métodos tradicionais, o melhor resultado foi obtido utilizando SVM com RGB + ExG + GLCM, com média de 0,78 de *F1-score* . Nas CNNs, a FRRN obteve o melhor resultado, com média de 0,80 de *F1-score*. Corroborando a ideia de que CNNs podem obter uma maior precisão na segmentação do que os métodos tradicionais.

Nos trabalhos Barbosa Júnior (2021), Barbosa Júnior et al. (2021), Barbosa Júnior et al. (2022) foram realizados experimentos em busca das principais caraterísticas ao se capturar imagens com VANTs de plantações de cana-de-açúcar para serem utilizar na detecção das linhas de plantio e/ou falhas. Uma análise exploratória foi realizada variando o GSD (3,5 cm, 6,0 cm e 8,2 cm), a altura da planta (0,5 m, 0,9 m, 1,2 m e 1,7 m) e o comprimento das falhas (0,5 m, 1,0 m, 1,5 m, 2,0 e 2,5 m) nas linhas de plantio. Na identificação das falhas foi utilizado o *software* comercial *Inforow* (INFOROW, 2021). Como esperado, os melhores resultados foram com GSD pequeno (3,5 cm ou próximo) e plantas com menor altura (e.g., 0,5 m). A altura das plantas foi apontada como um dos fatores mais importantes, seguido pelo GSD, para detecção das falhas nas plantações de cana-de-açúcar. Não foi possível detectar pequenas falhas ($\leq$ 1 m) quando a planta já tinha certa altura ($\geq$ 1 m) através do *software*. É importante notar que mesmo o *software* comercial não obteve bons resultados em certos cenários.

Em Rocha et al. (2022a) e Rocha et al. (2022b), foi utilizado o algoritmo KNN para identificar as regiões de interesse (as plantas ou cultura) de imagens de plantações de cana-de-açúcar obtidas por um VANT. Na sequência, o filtro de gradiente RGB foi utilizado para estimar a orientação das linhas de plantio. Em seguida, a binarização de imagens, operações morfológicas e modelos de geometria computacional são utilizados para detectar e mapear as linhas de plantio e suas falhas. Os autores afirmam que o método obteve bons resultados, com detecção das falhas próximos às obtidas manualmente, entretanto não utilizaram nenhuma métrica estatística de comparação.

Silva et al. (2022) desenvolveram uma abordagem para identificar falhas nas linhas de plantio utilizando uma variedade de métodos computacionais. Com primeira etapa

foi realizado o treinamento da PSPNet nos *datasets* Pereira Júnior e Wangenheim (2019) e Monteiro e Wangenheim (2019) de imagens de plantação de cana-de-açúcar capturadas por VANTs. O treinamento foi realizado por 200 épocas, com 60% dos dados para treinamento, 20% para validação e 20% para teste. Os autores afirmam terem obtido um *F1-score* de aproximadamente 0,92 nos testes. Na sequência os pixel identificados como ervas daninhas foram descartados, a imagem é convertida em tons de cinza e feito a binarização a partir do limiar obtido algoritmo de Otsu. Na próxima etapa é realizado a esqueletização da imagem e aplicada a TH para identificação das linhas. Por fim são realizados operações morfológicas para corrigir alguns erros na detecção das falhas nas linhas de plantio. Os autores não utilizaram nenhuma métrica estatística de comparação, apenas o *F1-score* na etapa de testes da PSPNet, e mesmo nessa etapa poucos detalhes foram descritos no trabalho

No trabalho de Oliveira et al. (2023) foi avaliado se a utilização de VANTs com a tecnologia *Real-time Kinematic* (RTK) poderia melhorar a precisão das falhas detectadas em imagens plantações de cana-de-açúcar. O RTK ajuda os sistemas de GPS a terem uma maior precisão na localização geográfica. As imagens foram processadas utilizando o *software* comercial *Agisoft Metashape*[4] (*Agisoft LLC*), versão *Professional*. Na sequência o pacote de código aberto *FIELDimageR*[5] (*OpenDroneMap*) para a linguagem R foi utilizado para identificar as falhas nas linhas de plantio. No *FIELDimageR* foi utilizado o IV VARI (*Visible Atmospherically Resistant Index*) com valor de -0,12 ou mais para as plantas e o oposto para solo/fundo. A utilização do RTK não afetou a identificação nem o comprimento das falhas detectadas. No entanto, o RTK propiciou uma maior precisão na determinação da localização geográfica das falhas.

Muitos trabalhos relacionados foram encontrados na literatura, contudo, uma boa parte deles utilizou imagens de baixíssimas altitudes, onde as linhas curvas se aproximam de linhas retas. Muitos utilizaram a Transformada de Hough ou Transformada de Radon para identificar ou refinar a identificação das linhas de plantio. Alguns trabalharam com imagens de plantações de cana-de-açúcar, mas é raro utilizarem imagens de variados estágios da cultura. Trabalhos recentes utilizam técnicas de DL, indicando serem promissoras nesse cenário e que ainda podem ser muito exploradas. Alguns trabalham utilizaram a U-Net e analisaram seus hiperparâmetros na tentativa de melhorar o desempenho da segmentação, entretanto não foi encontrado nenhum trabalho que analisasse o número de blocos e/ou o número de filtros em cada bloco. Além disso, e principalmente, não foi encontrado nenhum trabalho com foco em melhorar a detecção das linhas de plantio (para qualquer cultura) ajustando esses parâmetros nos modelos da U-Net.

[4] Disponível em: <https://www.agisoft.com/downloads/installer/>. Acesso em: 23 de março de 2023

[5] Disponível em: <https://github.com/OpenDroneMap/FIELDimageR>. Acesso em: 23 de março de 2023

Capítulo 4

Metodologia de pesquisa

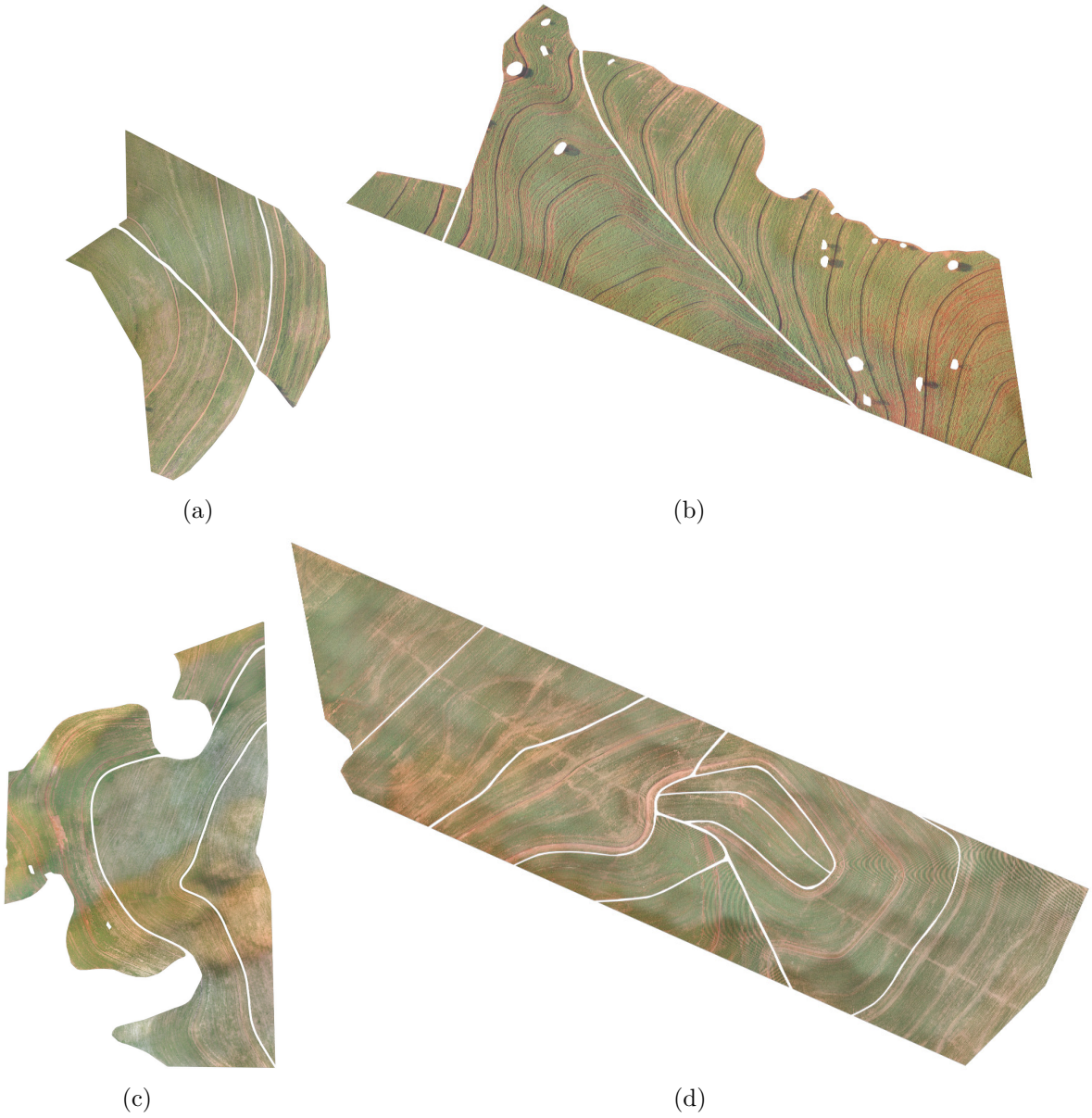
A metodologia deste projeto é composta por 4 etapas. A primeira etapa é a aquisição das imagens (ou *datasets*), por meio de VANTs utilizando uma câmera RGB em plantações de cana-de-açúcar. A segunda etapa é a segmentação das imagens, em que são testados e propostos métodos para melhorar a segmentação dessas imagens. Na sequência, é feita a avaliação dos resultados da detecção das linhas de plantio. Por fim, na quarta etapa é feito o pós-processamento de modo a otimizar a segmentação e refinar as linhas detectadas na etapa anterior.

4.1 Aquisição das Imagens

Para analisar a detecção de linhas de plantio em plantações de cana-de-açúcar, imagens de VANTs de *datasets* disponíveis online foram utilizadas. Em Silva, Escarpinati e Backes (2021), as imagens foram capturadas por um VANT de mapeamento, modelo *eBee SenseFly*, com uma câmera *SenseFly S.O.D.A.* de uma polegada, tendo 5472 × 3648 pixels de resolução e lente RGB F/2.8-11, 10,6 mm. Cada pixel na imagem representa aproximadamente 5 cm (GSD de 0,053 m).

As imagens foram capturadas de quatro diferentes plantações de cana-de-açúcar no Brasil, que após o processo de mosaicagem (processo de unir várias imagens individuais em uma única imagem maior) resultaram em quatro mosaicos (Figura 14). Os mosaicos foram nomeados de *dataset* A, B, C, e D, respectivamente. Eles contêm imagens de plantações canas-de-açúcar em variados estágios de cana-planta e de planta-soca. Cada mosaico teve suas linhas de plantio marcadas por um especialista, alguns exemplos podem ser visualizados na Figura 15. O especialista marcou com uma linha uniforme as linhas de plantio de cana-de-açúcar, onde a cultura está presente e onde ela deveria existir, ou seja, locais de falhas no plantio também foram marcados como linha de plantio.

A partir de uma análise visual dos *datasets*, pode-se afirmar que o *datasets* A, similar ao C, tem boa parte das linhas de plantio próximas de linhas retas. Por outro lado, os *datasets* B e D apresentam maior diversidade nas linhas de plantio, com maior variedade



(a) (b)

(c) (d)

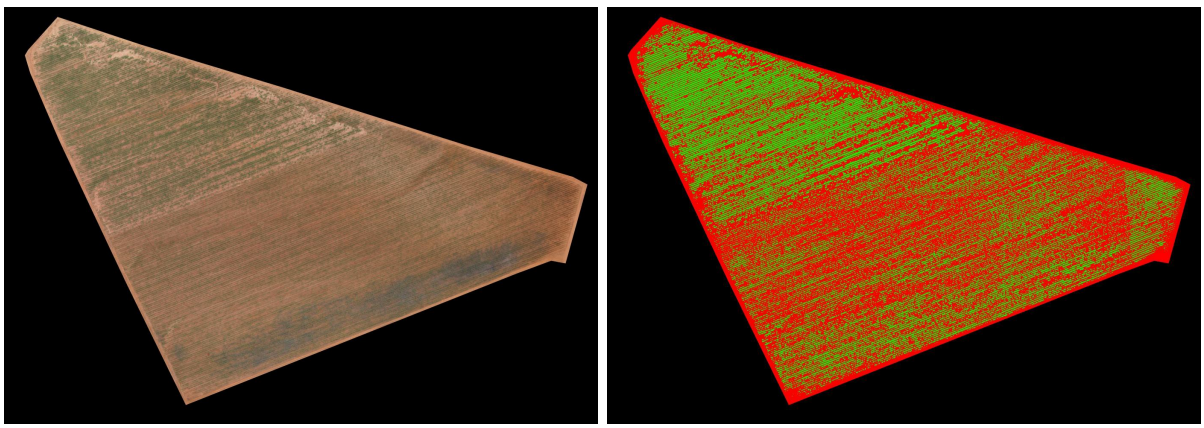
Figura 14 – Mosaicos e seus respectivos tamanhos: (a) *dataset* A, 11180 × 8449; (b) *dataset* B, 16677 × 24181; (c) *dataset* C, 17497 × 10771; (d) *dataset* D, 19833 × 30255.

Figura 15 – Exemplos de marcações realizadas pelo especialista nos mosaicos. A primeira coluna contém a imagem original, na segunda a marcação realizada pelo especialista e na terceira uma sobreposição das duas colunas anteriores, em vermelho a marcação do especialista.

de ângulos nos segmentos. O *dataset* A tem cor do solo com pouca variedade (i.e., próxima de um único tom), com pequenos locais de sombra e estágios semelhantes de crescimento da cana-de-açúcar, similar ao *dataset* C. Enquanto o *dataset* B contém mais locais com sombra e locais com falhas nas linhas de plantio, e também é possível visualizar mais estradas, que são utilizadas pelo maquinário agrícola.

Cada mosaico foi recortado em pedaços de 256 × 256 pixels sem sobreposição (i.e., *stride* de 256 pixels). Após o corte, foram descartadas amostras com menos de 80% de informação útil (pixels com valores diferentes de zero). Após o descarte, os *datasets* A, B, C, e D ficaram com total de 678, 3.291, 1.552 e 2.162 imagens, respectivamente.

Em Pereira Júnior et al. (2020a) e Pereira Júnior et al. (2020b), foi utilizado o mosaico "*Orthomosaic Dataset of RGB aerial Images for Crop Rows Detection*" (PEREIRA JÚNIOR; WANGENHEIM, 2019) do *Image Processing and Computer Graphics Lab* (LAPIX) da Universidade Federal de Santa Catarina (UFSC). Esse mosaico, denominado de *dataset* L neste projeto, foi capturado com uma câmera RGB *Canon G9X* e VANT *Horus Aeronaves*, resultando em um GSD aproximado de 5 cm. O mosaico, que pode ser visualizado na Figura 16, foi marcado por um especialista: as linhas de plantio em verde, o fundo/solo em vermelho e, fora do mosaico, todos os pixels foram definidos como pretos.



(a) (b)

Figura 16 – Mosaico L: (a) mosaico de tamanho, incluindo as bordas pretas, 6595 × 9391; (b) marcação do especialista (de mesmo tamanho), em verde as linhas de plantio, o fundo/solo em vermelho, e fora do mosaico como preto.

O mosaico "*Orthomosaic Dataset of RGB aerial Images for Weed Mapping*" (MONTEIRO; WANGENHEIM, 2019), também do LAPIX, poderia ser utilizado neste projeto, contudo a maior parte dele é composta apenas de solo/fundo e ervas daninhas, fugindo do foco deste projeto.

4.2 Segmentação

A próxima etapa é a segmentação, considerada a principal etapa de processamento, devido sua importância e complexidade. Inicialmente é fundamental conhecer as imagens que serão segmentadas e quais métodos de segmentação serão utilizados. Para este projeto foram utilizadas imagens de VANTs com as bandas RGB, por serem mais baratas de serem capturadas do que as com banda infravermelho e/ou infravermelho próximo. Essas últimas precisam de uma câmera específica que consiga capturar tais bandas. Contudo, algumas informações úteis poderiam ser obtidas através dessas bandas e IVs (como o NDVI). Como as imagens são de plantações de cana-de-açúcar, é certo encontrar nelas fileiras da cultura e variadas partes de solo exposto.

Sobre os métodos de segmentação, foram escolhidas três redes neurais (U-Net, LinkNet e PSPNet) para segmentar as imagens. As redes neurais têm melhor resultado na segmentação em comparação com os algoritmos tradicionais (e outros métodos, como AG), por utilizarem não apenas o tom avermelhado do solo e o esverdeado das plantas, mas outras características que conseguem extrair das imagens (SILVA, 2020). Dentre essas características, podem ser contornos, formatos de linhas e outros tipos de formatos específicos.

As redes para segmentação (apresentadas na Seção 2.2) são treinadas para segmentar as linhas de plantio utilizando algumas abordagens: com imagens (RGB) das plantações, Índices de Vegetação, ou a combinação de ambos. Dentre os IVs para imagens RGB, alguns são mais utilizados, como o ExG, entretanto vários outros podem ser utilizados, como pode ser visualizado na Tabela 1. A utilização dos IVs é motivada pela hipótese de que as redes apreenderão mais características visuais das plantas e do solo do que apenas com a imagem RGB, portanto capazes de diferenciá-los com maior precisão.

4.3 Pós-processamento

A etapa de pós-processamento é responsável por refinar a segmentação obtida na etapa anterior. Neste caso, a proposta é refinar as linhas detectadas pelas redes neurais, testando alguns métodos que podem ou não melhorar o resultado. As operações morfológicas são utilizadas para separar pequenos e finos filamentos ligando duas linhas de plantio (algum erro na segmentação, e.g., pela presença de ervas daninhas ou plantas caídas) ou agrupar pequenos filamentos da mesma linha de plantio (e.g., pela presença de pequenas falhas ou plantas deslocadas da sua linha de plantio). Após a aplicação das operações morfológicas, é esperado que essas pequenas imperfeições sejam corrigidas, deixando as imagens preparadas para as próximas operações.

4.4 Avaliação dos Resultados

Para a avaliação dos resultados é importante um método confiável e de fácil replicação/teste. Neste projeto foram comparadas as marcações realizadas pelos especialistas com as linhas de plantio detectadas pelos métodos propostos. Para comparar os resultados foi utilizado o Coeficiente de *Dice* (CD), também conhecido como Coeficiente de *Sørensen–Dice*, que é capaz de comparar o quão similar duas imagens binárias são. Considere A e B como imagens binárias, o CD é definido por (JIMENEZ; GONZALEZ; GELBUKH, 2016):

$$CD(A, B) = 2\frac{|A \cap B|}{|A| + |B|} \tag{9}$$

No resultado do CD, $0 \leq CD(A, B) \leq 1$, quanto mais próximo de 1, mais similares são as imagens A e B. Para a validação dos resultados foram utilizadas imagens de vários hectares de cana-de-açúcar contendo plantas de diferentes idades e tamanhos. Nessa validação, assume-se que o resultado gerado pelos especialistas (a marcação, de forma manual) é o melhor resultado esperado. Assim, quanto mais próximo desse resultado as imagens geradas pelos métodos propostos estiverem, mais assertivas elas são consideradas.

Nas CNNs foi utilizado o Coeficiente de Jaccard (CJ), que também é conhecido como *Intersection over Union* (IoU), como função de custo (*loss function*) na etapa de treinamento. Em bibliotecas de rede neural, o CJ geralmente já vem implementado, o que facilita a sua utilização e reduz possíveis erros de implementação. O CJ entre duas imagens A e B é definido por (JIMENEZ; GONZALEZ; GELBUKH, 2016):

$$CJ(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{10}$$

Semelhante ao CD, o CJ tem como resultado, $0 \leq CJ(A, B) \leq 1$, gradativamente mais próximo de 1 ao quanto as imagens A e B forem similares. É possível converter CD em CJ (Equação 11), e vice-versa (Equação 12) (TAHA; HANBURY, 2015):

$$CJ = \frac{CD}{2 - CD} \tag{11}$$

$$CD = \frac{2CJ}{1 + CJ} \tag{12}$$

Capítulo 5

Análise e discussão dos resultados

Neste capítulo são apresentados os experimentos desenvolvidos e suas análises nos quais o método proposto foi aplicado. Os experimentos foram executados em um computador pessoal com processador Intel® Core™ i7-7000 de 3,60 GHz, com 32 GiB de RAM, sistema operacional Windows 10 e placa de vídeo NVIDIA GeForce GTX 1050 Ti de 4 GiB GDDR5. Também foi utilizado o *Python* 3.9.7 com as bibliotecas (*Keras* 2.8.0, *Pandas* 1.4.1, *NumPy* 1.22.2, *Matplotlib* 3.5.1, *TensorFlow* 2.8.0, *OpenCV-Python* 4.5.5.62, *scikit-learn* 1.0.2, *scikit-image* 0.19.2) e o CUDA *Toolkit* 11.5.0 para o desenvolvimento e teste dos experimentos.

Os experimentos foram divididos em estudos de casos específicos. No primeiro, foi realizado uma análise de três CNNs na segmentação das imagens utilizando *transfer learning* e *fine-tuning*, com avaliação da influência de ambos no resultado (segmentação). O foco desse estudo de caso é encontrar qual das três redes consegue segmentar melhor as imagens do cenário avaliado.

Na sequência, a partir dos melhores resultados terem sido obtidos com a U-Net, foi realizado um estudo de caso avaliando o número de blocos e filtros convolucionais na U-Net e sua capacidade de segmentar as linhas de plantio. Esse estudo de caso avalia mudanças no modelo da U-Net que poderiam melhorar a segmentação das imagens e/ou reduzir o tempo e poder computacional necessário para utilização da rede.

Em seguida, a partir de duas das melhores configurações da U-Net do estudo de caso anterior, foi realizado uma análise buscando otimizar a segmentação das imagens utilizando Índices de Vegetação como dados complementares de entrada ou exclusivos.

No quarto estudo de caso, foram avaliados dois modelos da U-Net (treinados anteriormente) para realizar a segmentação das imagens do *dataset* L e também modelos treinados a partir do zero no *dataset* L. Esse estudo de caso busca analisar a capacidade dos modelos em segmentar imagens de outros *datasets* não utilizados no treinamento e com marcações realizadas de modo diferente pelos especialistas.

Por fim, no último estudo de caso foi analisado a possibilidade de melhorar os resultados (segmentação) obtidos utilizando operações morfológicas. Foram realizadas análises

utilizando quatro operações morfológicas e dois elementos estruturantes, a fim de avaliar se elas poderiam aprimorar a detecção das linhas de plantio previamente obtidas.

5.1 Estudo de caso de CNNs para segmentação com *transfer learning* e *fine-tuning*

Para analisar o *Deep Learning* na segmentação das linhas de plantio foram escolhidas três redes neurais (U-Net, LinkNet e PSPNet) muito utilizadas na segmentação semântica em aplicações da AP. O código das redes tem como base a biblioteca em Python *Segmentation Models*[1], na versão 1.0.1, de Iakubovskii (2019).

É importante ressaltar que devido às limitações da biblioteca *Segmentation Models*, as imagens de entradas nas redes devem ter um tamanho (altura e largura) específico. Para a U-Net e a LinkNet esses valores devem ser divisíveis por 32 (e.g., 256/32 = 8), portanto 256 × 256. Por outro lado, para a PSPNet esses valores devem ser divisíveis por $6 * downsample\_factor$ (taxa de *downsampling*). Essa taxa é a profundidade do *backbone* para construir o *pyramid pooling module* nele. Por padrão na biblioteca, essa taxa é 8 (podendo ser alterada para 4 ou 16), assim não foi possível definir as imagens de entrada para a PSPNet com o mesmo tamanho da U-Net e LinkNet (256 × 256), ficando com 240 × 240, (i.e., $5 * (6 * 8) = 240$).

Como *encoder*/*backbone* foi utilizada a rede VGG16 pré-treinada com o *dataset* ImageNet 2012. Seguindo a recomendação do autor do código, utilizou-se *fine-tuning* (ajuste fino), assim os parâmetros do *encoder* não são treináveis (*encoder_weights = 'imagenet', encoder_freeze = True*). A quantidade de parâmetros de cada rede neural pode ser visualizada na Tabela 2. A arquitetura das três redes pode ser visualizada na Tabela 3, em cinza as partes com origem na VGG16.

Tabela 2 – Quantidade de parâmetros de treinamento de cada rede utilizando como *backbone* a VGG16 pré-treinada com o *dataset* ImageNet 2012 e *fine-tuning*.

Rede	Total de parâmetros	Treináveis	Não treinável
U-Net	23.752.273	9.033.553	14.718.720
LinkNet	20.325.137	5.603.633	14.721.504
PSPNet	10.009.409	2.369.025	7.640.384

As imagens, antes do treinamento das redes neurais, são normalizadas em valores entre 0 e 1, através da Equação 13, onde o valor de cada banda RGB é divido por 255.

$$img\_normalizada = img\_entrada/float(255) \tag{13}$$

[1] Disponível em: <https://github.com/qubvel/segmentation_models>. Acesso em: 13 de maio de 2023.

Tabela 3 – Arquitetura dos três modelos utilizados (U-Net, LinkNet e PSPNet) com rede a VGG16 como *encoder*. As partes com fundo cinza são da rede VGG16.

U-Net	**LinkNet**	**PSPNet**
Entrada 256 × 256 × 3	Entrada 256 × 256 × 3	Entrada 240 × 240 × 3
2 Conv3-64	2 Conv3-64	2 Conv3-64
Max pooling 2	*Max pooling* 2	*Max pooling* 2
2 Conv3-128 – saída b2c2	2 Conv3-128 – saída b2c2	2 Conv3-128
Max pooling 2	*Max pooling* 2	*Max pooling* 2
3 Conv3-256 – saída b3c3	3 Conv3-256 – saída b3c3	3 Conv3-256
Max pooling 2	*Max pooling* 2	*Max pooling* 2
3 Conv3-512 – saída b4c3	3 Conv3-512 – saída b4c3	3 Conv3-512 – saída b4c3
Max pooling 2	*Max pooling* 2	Produz 5 saídas iguais
3 Conv3-512 – saída b5c3	3 Conv3-512 – saída b5c3	4 to AvgPooling(30, 15, 10, 5)
Max pooling 2	*Max pooling* 2	→ Conv1-512
2 (Conv3-512, B, ReLu)	2 (Conv3-512, B, ReLU)	→ B, ReLU
*UpSampling*2D	Conv1-128, B, ReLU	→ *UpSampling*2D
Concatenate b5c3	*UpSampling*2D	*Concatenate* os 4 com b4c3
2 (Conv3-256, B, ReLU)	Conv3-128, B, ReLU	Conv1-512
*UpSampling*2D	Conv1-512, B, ReLU	B, ReLU
Concatenate b4c3	*Add* b5c3	Conv3-1
2 (Conv3-128, B, ReLu)	Conv1-128, B, ReLU	*UpSampling*2D
*UpSampling*2D	*UpSampling*2D	*Sigmoid*
Concatenate b3c3	Conv3-128, B, ReLU	Saída 240 × 240 × 1
2 (Conv3-64, B, ReLU)	Conv1-512, B, ReLU	
*UpSampling*2D	*Add* b4c3	
Concatenate b2c2	Conv1-128, B, ReLU	
2 (Conv3-32, B, ReLu)	*UpSampling*2D	
*UpSampling*2D	Conv3-128, B, ReLU	
2 (Conv3-16, B, ReLU)	Conv1-256, B, ReLU	
Conv3-1	*Add* b3c3	
Sigmoid	Conv1-64, B, ReLU	
Saída 256 × 256 × 1	*UpSampling*2D	
	Conv3-64, B, ReLU	
	Conv1-128, B, ReLU	
	Add b2c2	
	Conv1-32, B, ReLU	
	*UpSampling*2D	
	Conv3-32, B, ReLU	
	Conv1-16, B, ReLU	
	Conv3-1	
	Sigmoid	
	Saída 256 × 256 × 1	

Conv<tamanho do filtro>-<quantidade de canais>, por exemplo: Conv3-64, foi realizado a operação de convolução 3 × 3 com 64 canais.
B – *Batch Normalization* (normalização em lote).
bXcY – resultado do bloco X após a convolução Y.

As redes neurais foram treinadas por 50 épocas (*epochs*) utilizando o otimizador Adam com 0,001 de taxa de aprendizado (*learning rate*) e tamanho de lote (*batch size*) de 2. Como função de custo e métrica, foi utilizado o CJ. Alguns testes foram realizados utilizando como função de custo e métrica o CD, contudo os resultados foram muito similares aos utilizando CJ, assim o último foi escolhido para ser utilizado, de modo a evitar erros na implementação e ser o mais utilizado na literatura. Também foram utilizados métodos simples de aumento de dados (*data augmentation*) - como rotação (até ±180°), translação (até ±20px), dimensionamento (até ±7%) e cisalhamento - durante o processo de treinamento de modo a evitar o *overfitting*.

De modo a garantir o correto processo de validação, foi utilizado a validação cruzada *k-fold*, com $k = 10$. Deste modo, os dados foram divididos em 10 subconjuntos mutuamente exclusivos, onde um subconjunto (10% dos dados) é utilizado para teste e os outros nove para treinamento (90% dos dados). O subconjunto de teste foi alternado de forma circular até completar os 10 treinamentos de cada rede.

O treinamento das três redes para cada um dos quatro *datasets* gerou 12 combinações diferentes e pela utilização do *k-fold* com $k = 10$, resultou em 120 modelos diferentes. Para cada modelo foi realizada a predição do *dataset* utilizado no treinamento, salvando os resultados (segmentação) obtidos. Em seguida, o Coeficiente de *Dice* (CD) de cada segmentação foi calculado em relação à marcação realizada pelo especialista e no final é calculado a média do CD para todo o *dataset* analisado no momento. Por fim, os valores foram agrupados de acordo com cada uma das 12 combinações, calculando a média e desvio padrão dos 10 resultados de cada combinação.

Na Tabela 4 pode ser visualizado a média do CD e o respectivo desvio padrão obtido em cada combinação (treinamentos e teste no mesmo *dataset*). Para todas as redes, os melhores resultados foram obtidos no *dataset* B, enquanto utilizando o *dataset* D foram obtidos os piores resultados. A U-Net obteve os melhores resultados (em negrito) para todos os *datasets*. É importante notar que a variação (desvio padrão) nos resultados é bem pequena, de 0,0081 no pior caso - na PSPNet que obteve os piores resultados, exceto no *dataset* D.

Tabela 4 – Resultados do CD médio e desvio padrão obtidos para cada rede no *dataset* utilizado no treinamento, com *k-fold* ($k = 10$). Em negrito os melhores resultados para cada *dataset*.

Rede	***Dataset***			
	A	B	C	D
U-Net	**0,9080** ± 0,0045	**0,9233** ± 0,0031	**0,9044** ± 0,0028	**0,8672** ± 0,0056
LinkNet	0,8907 ± 0,0029	0,9030 ± 0,0024	0,8835 ± 0,0024	0,8358 ± 0,0056
PSPNet	0,8707 ± 0,0075	0,8861 ± 0,0081	0,8661 ± 0,0065	0,8410 ± 0,0079

Como próxima etapa, foi analisada a habilidade das redes de generalizar as caraterísticas extraídas para segmentar diferentes *datasets*. Utilizando os 120 modelos treinados

anteriormente, foi realizada a predição para os demais *datasets* não utilizados no treinamento e depois o cálculo CD e sua média. Na Tabela 5 pode ser analisado o CD médio obtido em cada combinação de rede, treinamento e teste. De modo a facilitar a análise dos resultados, o desvio padrão não foi adicionado nessa tabela, mas pode ser visualizado na Tabela 30 do Apêndice A.

Tabela 5 – Resultados do CD médio obtidos para cada rede durante o treinamento e teste nos *datasets*. Em negrito treinamento e teste no mesmo *dataset*.

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	A	**0,9080**	0,8851	0,8494	0,8150
	B	0,8181	**0,9233**	0,8922	0,7440
	C	0,6322	0,8847	**0,9044**	0,5288
	D	0,8914	0,8938	0,8634	**0,8672**
LinkNet	A	**0,8907**	0,8585	0,8311	0,7905
	B	0,7969	**0,9030**	0,8688	0,7284
	C	0,7133	0,8665	**0,8835**	0,6195
	D	0,8743	0,8653	0,8313	**0,8358**
PSPNet	A	**0,8707**	0,7982	0,7393	0,7596
	B	0,7622	**0,8861**	0,7737	0,7599
	C	0,5999	0,8420	**0,8661**	0,6046
	D	0,8587	0,6646	0,5487	**0,8410**

Os melhores resultados são esperados ao se utilizar o mesmo *dataset* no treinamento e no teste, representando a diagonal principal (em negrito) de cada rede na Tabela 5. Deste modo, a segunda linha da U-Net significa que ela foi treinada com o *dataset* B, obtendo média de CD de 0,9233 e nos testes em A, C e D, respectivamente 0,8181, 0,8922 e 0,7440. Apesar da utilização de técnicas de aumento de dados, no *dataset* D os melhores resultados foram obtidos nos testes realizados em outros *datasets*, e não no próprio *dataset*. Por outro lado, ao fazer o treinamento no *dataset* C e testar nos outros *datasets*, foram obtidos resultados muito inferiores aos demais, principalmente no *dataset* A e D, e um maior desvio padrão.

De modo geral, as redes tiveram dificuldade para generalizar as características apreendidas de um *dataset* para os outros. Apesar de o *dataset* A ter a menor quantidade de imagens (678), foram os *datasets* C e D (com 1.552 e 2.162 imagens respectivamente) que tiveram maiores oscilações nos resultados, muitas vezes apresentando um resultado inferior aos demais. Uma possível explicação para essa dificuldade em generalizar as características apreendidas está no fato dos *datasets* terem imagens de cana-de-açúcar em estágios de cana-planta e cana-soca em variados estágios de crescimento. A cana-planta apresenta um solo mais exposto e um maior contraste entre o esverdeado das plantas e o aspecto avermelhado do solo. Por outro lado, na cana-soca é provável que existam muitas folhas secas e sobras da planta no solo do último corte, dificultando a análise computacio-

nal, já que essas regiões podem ser confundidas como parte da linha plantio. Um exemplo de cana-planta versus cana-soca pode ser visualizado na Figura 13.

Com base nos resultados anteriores e de que o aumento de dados não conseguiu resolver o problema das variações nos resultados entre os *datasets*, uma nova abordagem foi criar um *dataset* com imagens dos quatro *datasets* utilizados. O novo *dataset* foi nomeado de E. De modo a também avaliar o impacto da quantidade de imagens, esse *dataset* precisa ter N, N = {200, 300, 400, 500}, imagens selecionadas aleatoriamente de cada um dos outros *datasets*. Deste modo, E(200) possui 800 imagens enquanto E(500) possui 2.000 imagens. As técnicas de aumento de dados também foram utilizadas com a mesma configuração dos experimentos anteriores. Na Tabela 6 podem ser analisados os resultados obtidos ao treinar as redes nos *datasets* E(N) e seu teste nos demais *datasets*, o desvio padrão pode ser visualizado na Tabela 31 no Apêndice A.

Tabela 6 – Resultados do CD médio obtidos ao treinar cada rede nos *datasets* E(N) e testar nos outros *datasets*.

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	E(200)	0,9049	0,9099	0,8916	0,8422
	E(300)	0,9043	0,9104	0,8955	0,8482
	E(400)	0,9069	0,9137	0,8968	0,8495
	E(500)	0,9096	0,9155	0,8993	0,8547
LinkNet	E(200)	0,8821	0,8905	0,8697	0,8090
	E(300)	0,8835	0,8920	0,8740	0,8206
	E(400)	0,8823	0,8936	0,8742	0,8196
	E(500)	0,8849	0,8952	0,8762	0,8238
PSPNet	E(200)	0,8653	0,8679	0,8560	0,8158
	E(300)	0,8648	0,8713	0,8523	0,8185
	E(400)	0,8751	0,8779	0,8605	0,8263
	E(500)	0,8748	0,8795	0,8609	0,8304

Combinar imagens de diferentes *datasets* melhorou significativamente os resultados obtidos e assim a capacidade das redes neurais de segmentar as imagens e de generalizar as características apreendidas. Além disso, a quantidade de imagens nos *datasets* E(N) teve pouca importância no desempenho das redes, sendo um pouco mais notável na PSPNet.

Nos resultados anteriores (Tabela 5), o desempenho das redes mudou drasticamente ao testar nos outros *datasets* que não foram usados no treinamento. Por outro lado, treinar as redes utilizando as combinações aleatórias de imagens (*datasets* E(N)) proporcionou melhores resultados para os 4 *datasets* (i.e., A, B, C, D), principalmente para a PSPNet. Os resultados dos testes nos *datasets* E(N) (treinamento e teste nos *datasets* E(N)) não foram considerados por não serem muito relevantes para esse experimento e, devido a sua natureza aleatória, terem sua forma praticamente desconhecida para o leitor.

Nesse resultado, como nos anteriores, a U-Net apresentou os melhores resultados na segmentação das imagens, o que é notável, já que a U-Net é considerada mais simples que

a LinkNet e a PSPNet. Contudo, é importante ressaltar que as imagens de entrada para a PSPNet tiveram que ser reduzidas, de 256 × 256 para 240 × 240, o que talvez reduziu um pouco sua capacidade de aprendizado e assim influenciando nos seus resultados.

5.1.1 Influência do *transfer learning* e *fine-tuning* na segmentação

A partir dos resultados anteriores (Estudo de caso 1), foi levantada a questão da influência da utilização do *transfer learning* (transferência de aprendizado) e do *fine-tuning* nos resultados. Assim, dois novos experimentos foram planejados. O primeiro (Estudo de caso 2), ainda utilizando o *transfer learning*, mas sem congelar os parâmetros do *encoder* (*encoder_weights = 'imagenet', encoder_freeze = False*). Já o segundo (Estudo de caso 3), sem utilizar *transfer learning* e por conseguinte sem *fine-tuning* (*encoder_weights = 'none', encoder_freeze = False*).

A quantidade de parâmetros é a mesma para os dois novos experimentos e pode ser visualizada na Tabela 7. Comparado com o experimento anterior, uma maior quantidade de parâmetros está disponível para treinamento. Para agilizar os testes, foi utilizado 90% das imagens para treinamento e 10% para teste, mas sem utilizar a estratégia *k-fold* utilizada anteriormente. Os outros parâmetros são os mesmos (i.e., treinamento por 50 épocas utilizando otimizador Adam com 0,001 de taxa de aprendizado, tamanho de lote de 2 e os mesmos métodos simples de aumento de dados).

Tabela 7 – Quantidade de parâmetros de treinamento para os Estudos de caso 2 e 3.

Rede	Total de parâmetros	Treináveis	Não treináveis
U-Net	23.752.273	23.748.241	4.032
LinkNet	20.325.137	20.318.321	6.816
PSPNet	10.009.409	10.004.289	5.120

Não congelar os parâmetros do *encoder*, deixando livre os parâmetros do *backbone* para treinamento, melhorou os resultados em cerca de 3% (Tabela 8). A abordagem de combinar imagens dos outros *datasets* também proporcionou uma melhor estabilidade na rede ao ser testada nos outros *datasets* (Tabela 9). A PSPNet apresentou alguns resultados bem inferiores, no *dataset* B e no E(300), que talvez possam ser explicado por um treinamento insuficiente (*underfitting*).

Outro ponto importante a ser analisado é tempo de treinamento de cada rede. Apesar do melhor resultado obtido ao não congelar os parâmetros do *encoder*, mais parâmetros ficaram disponíveis para treinamento, assim aumentando o tempo necessário para o treinamento das redes. Como pode ser observado na Tabela 10, o tempo médio, em segundos, necessários para treinar cada uma das 50 épocas das redes neurais aumentou em cerca de 100%. A LinkNet, comparada com a U-Net, possui um tempo de treinamento cerca de 5

Tabela 8 – Resultados do CD médio obtidos utilizando *transfer learning*, mas sem congelar os parâmetros do *encoder*.

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	A	**0,9196**	0,9101	0,8946	0,8389
	B	0,8848	**0,9367**	0,9102	0,8587
	C	0,8875	0,9288	**0,9221**	0,8722
	D	0,8894	0,9106	0,9087	**0,8953**
LinkNet	A	**0,9183**	0,9056	0,8890	0,8395
	B	0,9056	**0,9385**	0,9144	0,8686
	C	0,8610	0,9167	**0,9150**	0,8259
	D	0,8998	0,9140	0,9071	**0,8922**
PSPNet	A	**0,9083**	0,8991	0,8843	0,8409
	B	0,5864	**0,6123**	0,5936	0,5996
	C	0,8815	0,9127	**0,9093**	0,8605
	D	0,8993	0,8935	0,8849	**0,8509**

Tabela 9 – Resultados do CD médio obtidos ao treinar cada rede nos *datasets* E(N) e testar nos outros *datasets* utilizando *transfer learning*, mas sem congelar os parâmetros do *encoder*.

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	E(200)	0,9172	0,9147	0,8919	0,8412
	E(300)	0,9204	0,9261	0,9146	0,8771
	E(400)	0,9121	0,9297	0,9125	0,8772
	E(500)	0,9183	0,9321	0,9147	0,8771
LinkNet	E(200)	0,9240	0,9232	0,9052	0,8594
	E(300)	0,9202	0,9225	0,9094	0,8683
	E(400)	0,9127	0,9235	0,8982	0,8529
	E(500)	0,9132	0,9298	0,9160	0,8742
PSPNet	E(200)	0,9171	0,9159	0,9007	0,8575
	E(300)	0,5864	0,6123	0,5936	0,5996
	E(400)	0,9169	0,9113	0,8928	0,8544
	E(500)	0,9100	0,9147	0,9027	0,8663

segundos menor. Por outro lado, a PSPNet tem tempos menores, mas aumentaram em mais de 100% em alguns casos.

Tabela 10 – Tempo médio de treinamento de cada época (em segundos) de cada rede neural.

Rede	*Dataset*	Estudo de caso 1	Estudo de caso 2 e 3
U-Net	A	39	59
	B	152	275
	C	73	132
	D	102	183
	E(200)	44	70
	E(300)	57	104
	E(400)	75	136
	E(500)	94	170
LinkNet	A	35	55
	B	143	269
	C	68	128
	D	95	178
	E(200)	38	65
	E(300)	53	99
	E(400)	70	131
	E(500)	88	165
PSPNet	A	20	36
	B	79	171
	C	38	82
	D	53	111
	E(200)	23	43
	E(300)	29	62
	E(400)	38	83
	E(500)	47	104

Sem a utilização do *transfer learning*, e por consequência o *fine-tuning* (Estudo de caso 3), os resultados para a U-Net e LinkNet ficam bem próximos para o primeiro caso (Estudo de caso 2), como pode ser visualizado na Tabela 11. Contudo, os resultados para a PSPNet foram muito inferiores, provavelmente pelo treinamento insuficiente considerando os hiperparâmetros utilizados nos experimentos (e.g., 50 épocas, otimizador Adam com taxa de aprendizado de 0,001). Mesmo a abordagem de combinar imagens do outros *datasets* não foi suficiente para melhorar os resultados da PSPNet, obtendo a mesma média de CD independente do *dataset* utilizado no treinamento. Essa baixa média no CD representa uma falha no aprendizado das características das imagens pela rede. A PSPNet teve como resultado da segmentação imagens totalmente brancas ou totalmente pretas, i.e., a rede segmentou a imagem como sendo totalmente composta de cana-de-açúcar ou de solo.

Em resumo, a U-Net e a LinkNet obtiveram os melhores resultados, em qualquer um dos 3 cenários. A U-Net foi a rede com melhor desempenho, apesar de ter obtido

Tabela 11 – Resultados do CD médio obtidos sem utilizar *transfer learning* (treinamento a partir do zero).

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	A	**0,9186**	0,9042	0,8921	0,8379
	B	0,9151	**0,9380**	0,9140	0,8675
	C	0,8765	0,9066	**0,9150**	0,8539
	D	0,8908	0,9114	0,9059	**0,8847**
LinkNet	A	**0,9217**	0,9146	0,8896	0,8441
	B	0,9104	**0,9367**	0,9109	0,8605
	C	0,8893	0,9206	**0,9170**	0,8457
	D	0,8997	0,9111	0,9029	**0,8759**
PSPNet	A	**0,5864**	0,6123	0,5936	0,5996
	B	0,5864	**0,6123**	0,5936	0,5996
	C	0,5864	0,6123	**0,5936**	0,5996
	D	0,5864	0,6123	0,5936	**0,5996**

Tabela 12 – Resultados do CD médio obtidos ao treinar cada rede nos *datasets* E(N) e testar nos outros *datasets* sem utilizar *transfer learning*.

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	E(200)	0,9180	0,9205	0,8950	0,8507
	E(300)	0,9268	0,9146	0,8971	0,8548
	E(400)	0,9109	0,9187	0,8888	0,8378
	E(500)	0,9161	0,9321	0,9192	0,8799
LinkNet	E(200)	0,9160	0,9147	0,8922	0,8449
	E(300)	0,9250	0,9243	0,9107	0,8640
	E(400)	0,9184	0,9294	0,9135	0,8709
	E(500)	0,9156	0,9283	0,9174	0,8748
PSPNet	E(200)	0,5864	0,6123	0,5936	0,5996
	E(300)	0,5864	0,6123	0,5936	0,5996
	E(400)	0,5864	0,6123	0,5936	0,5996
	E(500)	0,5864	0,6123	0,5936	0,5996

alguns resultados ligeiramente inferiores que os da LinkNet. O *transfer learning* foi útil para o cenário analisado, principalmente para redes neurais com menor quantidade de parâmetros e para *datasets* com menor quantidade de imagens. O *fine-tuning* reduziu um pouco o desempenho das redes, provavelmente devido ao ImageNet ser bem diferentes das imagens utilizadas neste projeto. Por outro lado, diminuiu para próximo da metade o tempo necessário para o treinamento das redes, o que pode ser interessante para locais com recurso computacional limitado ou com custo por hora de processamento alto.

5.2 Estudo de caso da U-Net e o número de blocos e filtros convolucionais

Nos resultados obtidos no experimento anterior, a U-Net obteve, na maioria dos casos, os melhores resultados ao segmentar as imagens de cana-de-açúcar capturadas por VANTs. Assim, foi a rede neural escolhida para ser analisada com maior riqueza de detalhes. Este experimento tem como objetivo analisar a influência da quantidade de blocos convolucionais e de filtros nos resultados da segmentação da U-Net. Diferente do experimento anterior, aqui a rede foi implementada do zero de modo a permitir e facilitar a alteração na quantidade de blocos e filtros utilizados. As configurações avaliadas podem ser analisadas na Tabela 13.

Tabela 13 – Configurações da U-Net utilizadas nos experimentos.

Configuração	Quantidade de blocos e filtros	Total de parâmetros	Porcentagem
01	[16, 32, 64, 128, 256]	3.331.697	100,00%
02	[16, 32, 64, 128]	823.921	24,72%
03	[16, 32, 64]	196.721	5,90%
04	[16, 32]	39.793	1,19%
05	[16]	2.785	0,08%
06	[16, 16]	14.369	0,43%
07	[16, 16, 16]	25.953	0,77%
08	[16, 16, 16, 16]	37.537	1,12%
09	[16, 16, 16, 16, 16]	49.121	1,47%
10	[32]	10.177	0,30%
11	[64]	38.785	1,16%
12	[128]	151.297	4,54%

A Configuração 01 (i.e., [16, 32, 64, 128, 256]) foi utilizada como configuração padrão nas comparações, ou seja, a configuração na qual se encontra maior número de filtros por bloco/camada e por consequência maior número de parâmetros para treinamento e assim é esperado obter os melhores resultados. Ela possui 5 blocos/camadas de filtros, iniciando com 16 filtros convolucionais, duplicando a quantidade de filtros a cada novo bloco, assim como nas outras configurações. Essa configuração difere da configuração padrão da U-Net (RONNEBERGER; FISCHER; BROX, 2015), que inicia com 64 filtros, duplicando até completar o quinto e último bloco com 1.024 filtros.

A U-Net utiliza os blocos de filtros em ordem direta na primeira parte (*contraction path*) e na ordem inversa na segunda parte (*expansive path*), embora o último bloco (e.g., 1.024 da configuração padrão) não seja usado diretamente no *expansive path*. Cada bloco de filtros é utilizado duas vezes nas convoluções 3×3 (seta azul na Figura 6).

Para cada configuração na Tabela 13, a U-Net foi treinada para os quatro *datasets* (i.e., A, B, C e D), como entrada uma imagem de $256 \times 256 \times 3$, normalizada entre

0 e 1, como no estudo apresentado na seção 5.1 (através da Equação 13). De modo a agilizar os testes, foram utilizados 80% das imagens para treinamento e 20% para teste ao invés da estratégia *k-fold* utilizada anteriormente. Também não foi utilizado *transfer learning* e nenhum aumento de dados. A rede foi treinada por 50 épocas, utilizando o otimizador Adam com taxa de aprendizado de 0,001 e tamanho de lote de 8. Os filtros são de tamanho 3×3, utilizando *padding same*, *strides* de 1, função de custo a *binary cross-entropy* e métrica o CJ.

Após o treinamento, para cada configuração, foi realizada a predição (segmentação das imagens de cana-de-açúcar) de cada um dos *datasets*. A partir das predições foi calculado a média do CD para cada *dataset*, os resultados podem ser analisados na Figura 17. Eles estão organizados em forma de matriz, onde a linha representa o *dataset* utilizado no treinamento e nas colunas o *dataset* utilizado na predição (e.g., B×C na Configuração 01, representa que a U-Net foi treinada no *dataset* B utilizando a Configuração 01 e depois realizada a predição do *dataset* C, resultando em 0,90 de CD em média).

Configuração 01: [16, 32, 64, 128, 256]

	A	B	C	D
A	0,96	0,68	0,84	0,73
B	0,90	0,97	0,90	0,84
C	0,90	0,91	0,96	0,86
D	0,87	0,91	0,89	0,94

Configuração 02: [16, 32, 64, 128]

	A	B	C	D
A	0,94	0,69	0,82	0,70
B	0,88	0,95	0,89	0,80
C	0,87	0,91	0,94	0,79
D	0,84	0,87	0,85	0,92

Configuração 03: [16, 32, 64]

	A	B	C	D
A	0,92	0,69	0,81	0,69
B	0,64	0,93	0,85	0,65
C	0,61	0,88	0,90	0,49
D	0,83	0,86	0,81	0,88

Configuração 04: [16, 32]

	A	B	C	D
A	0,86	0,60	0,74	0,65
B	0,54	0,89	0,75	0,57
C	0,65	0,79	0,85	0,57
D	0,78	0,79	0,74	0,81

Configuração 05: [16]

	A	B	C	D
A	0,69	0,14	0,42	0,59
B	0,27	0,69	0,61	0,24
C	0,50	0,60	0,65	0,41
D	0,64	0,30	0,49	0,63

Configuração 06: [16, 16]

	A	B	C	D
A	0,84	0,53	0,74	0,63
B	0,55	0,88	0,73	0,56
C	0,69	0,82	0,85	0,61
D	0,77	0,76	0,72	0,80

Configuração 07: [16, 16, 16]

	A	B	C	D
A	0,90	0,67	0,79	0,69
B	0,69	0,92	0,86	0,67
C	0,70	0,87	0,88	0,52
D	0,77	0,81	0,80	0,85

Configuração 08: [16, 16, 16, 16]

	A	B	C	D
A	0,92	0,63	0,82	0,71
B	0,84	0,94	0,89	0,76
C	0,84	0,89	0,92	0,70
D	0,85	0,89	0,85	0,89

Configuração 09: [16, 16, 16, 16, 16]

	A	B	C	D
A	0,93	0,67	0,83	0,71
B	0,91	0,94	0,91	0,83
C	0,91	0,91	0,92	0,80
D	0,84	0,90	0,86	0,90

Configuração 10: [32]

	A	B	C	D
A	0,75	0,25	0,48	0,62
B	0,29	0,71	0,61	0,28
C	0,51	0,61	0,67	0,43
D	0,65	0,34	0,50	0,64

Configuração 11: [64]

	A	B	C	D
A	0,76	0,38	0,54	0,63
B	0,36	0,75	0,63	0,37
C	0,54	0,63	0,69	0,45
D	0,62	0,24	0,50	0,63

Configuração 12: [128]

	A	B	C	D
A	0,74	0,29	0,47	0,59
B	0,26	0,71	0,62	0,25
C	0,49	0,60	0,68	0,40
D	0,69	0,39	0,57	0,68

Figura 17 – Resultados do CD médio obtidos para as 12 configurações na averiguação da influência do número de filtros e blocos convolucionais na segmentação com a U-Net. Para cada teste, a linha identifica o *dataset* utilizado no treinamento, enquanto as colunas os *datasets* preditos/avaliados. Os tons em vermelho destacam a deterioração do CD em relação ao resultado na diagonal da Configuração 01 (em azul).

A diagonal em cada matriz de resultado representa que o treinamento e teste foi executado no mesmo *dataset*. Ou seja, após o treinamento com 80% da base, a rede foi utilizada para fazer a predição de todo o *dataset*. Assim é esperado ter os melhores resultados de cada matriz. As marcações em vermelho destacam a deterioração do CD

em relação ao resultado na diagonal da Configuração 01 (em azul).

A redução da quantidade de blocos convolucionais de 5 para 4 e 3 não degradou muito o resultado, exceto em alguns casos específicos em que o treinamento e o teste foram realizados em *datasets* diferentes, como na Configuração 03 C × D. Nesses casos, o *dataset* utilizado no treinamento pode não ter fornecido as características necessárias (e.g., variedade de solo exposto e estágios de crescimento) para identificar adequadamente as linhas de plantio nas imagens do outro *dataset*. É importante notar que essa redução resultou em uma diminuição de mais de 94% (Configuração 03) no número de parâmetros.

Por outro lado, a redução para 2 ou 1 bloco convolucional degradou muito os resultados, inclusive quando avaliado no próprio *dataset* de treino (diagonal principal). A Configuração 05 tem apenas 2.785 parâmetros, ou 0,08% da configuração de comparação (Configuração 01), e teve uma grande degradação no resultado. Essa degradação presumivelmente foi pela significativa redução no número de parâmetros (treináveis) ou, ainda mais provável, pela redução do número de blocos convolucionais para apenas um, ou pela combinação de ambos.

É importante ressaltar que ao utilizar apenas um bloco convolucional, a U-Net não utiliza as *skip conections* e, por conseguinte, a concatenação do resultado dos blocos convolucionais anteriores (como pode ser observado na Figura 18). Dessa forma, não é plenamente explorado o potencial das informações locais e globais presentes nas imagens.

Para avaliar a hipótese mais provável, foram avaliadas configurações da U-Net utilizando um único bloco com diferentes números de filtros (Configuração 10, 11 e 12), e diferentes números de blocos utilizando a mesma quantidade de filtros por cada bloco (Configuração 06, 07, 08 e 09). A partir dos resultados, é possível averiguar que apenas um bloco convolucional, com 16, 32, 64 ou até 128 filtros, não conseguiu apreender as características das imagens para segmentá-las adequadamente.

Aumentar a quantidade de blocos, em contrapartida, mesmo com poucos filtros em cada um, produziu resultados próximos da configuração da U-Net com a maior quantidade de parâmetros. A configuração com 5 blocos convolucionais de 16 filtros cada (Configuração 09 - 49.121 parâmetros) contém menos de 3 vezes a quantidade de parâmetros do que a configuração com apenas um bloco convolucional de 128 filtros (Configuração 12 - 151.297 parâmetros), mas apresenta resultados significativamente melhores.

Cada bloco adicionado nas Configurações 05 até 09 resultou em perceptível melhoria nos resultados. Os resultados da Configuração 09 foram próximos da Configuração 01, utilizando apenas 1,47% dos parâmetros. Esses resultados corroboram a hipótese de que o número de blocos convolucionais está diretamente relacionado com a capacidade de aprendizado das características semânticas pela rede neural. Assim, o número de blocos convolucionais tem maior influência no resultado do que o total de parâmetros da rede neural, pelo menos no cenário analisado com a U-Net.

Outro ponto observado, como no estudo apresentado na seção 5.1, foi a dificuldade em

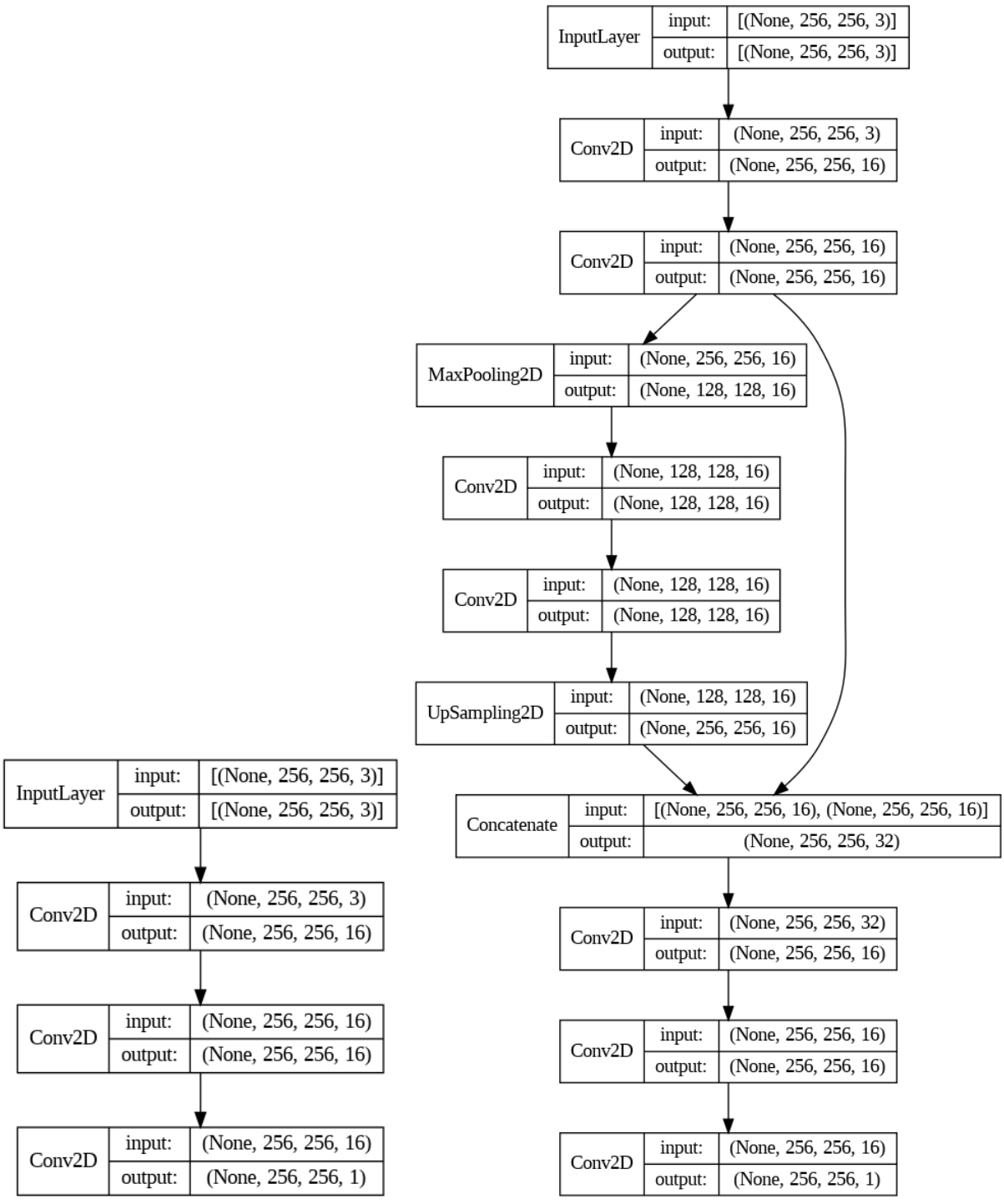


Figura 18 – Comparação entre a Configurações 05 e 06 da U-Net: (a) Configuração 05 ([16]), não utiliza *skip conections* e concatenação; (b) Configuração 06 ([16, 16]), utiliza *skip conections* e concatenação.

generalizar as características semânticas apreendidas em *datasets* diferentes do utilizado no treinamento. Esse fenômeno é mais perceptível ao efetuar a predição nos *datasets* B e D após o treinamento no *dataset* A. Até mesmo as configurações que obtiveram os melhores resultados (Configuração 01 e 09), apresentaram um desempenho inferior quando comparado com o obtido utilizando outros *datasets* para treinamento.

O *dataset* A tem a menor quantidade de imagens e, presumivelmente, a menor diversidade de imagens (i.e., imagens de plantações de cana-de-açúcar nos variados estágios de crescimento e solo exposto). Posto isto, é uma provável explicação do seu peculiar desempenho. Para resolver esse problema, usualmente se utiliza a estratégia de aumento de dados. Contudo, as variedades de solos, estágios de crescimentos das plantas e formatos das linhas de plantios não podem ser reproduzidos por simples técnicas de aumentos de dados (e.g., rotação, translação, mudança de escala e recorte). Assim, para resolver esses problemas, foi utilizado a estratégia de criar um *dataset* com a combinação de imagens do outros.

O novo *dataset*, nomeado de E(500), contém 500 imagens selecionadas aleatoriamente de cada um dos outros 4 *datasets*, totalizando 2.000 imagens. Foi realizado o treinamento nesse *dataset* e na sequência a predição e cálculo do CD para os outros *datasets*, como pode ser analisado na Tabela 14.

Tabela 14 – Configuração 01 e 09 realizadas no *dataset* E(500).

Configuração 01: [16, 32, 64, 128, 256]					**Configuração 09**: [16, 16, 16, 16, 16]				
	A	B	C	D		A	B	C	D
E(500)	0,96	0,94	0,94	0,87	E(500)	0,94	0,93	0,92	0,86

A partir dos resultados é possível afirmar que a combinação aleatória de imagens propiciou uma melhora na diversidade das imagens para o treinamento da rede. Uma maior diversidade possibilitou a U-Net apreender e a generalizar melhor as características semânticas através dos diferentes *datasets*. Desse modo, melhorou seu desempenho até mesmo nos *datasets* B e D, os quais havia apresentado o pior desempenho. É importante ressaltar que isso foi possível sem a necessidade de *transfer learning* e aumento de dados, apenas com o aumento da diversidade de imagens.

A título de comparação, em Silva, Escarpinati e Backes (2021), utilizando métodos tradicionais e nos mesmos *datasets* (i.e., *dataset* A, B, C e D), os melhores resultados obtidos ficam abaixo de 0,77 de CD. Por outro lado, a abordagem utilizando a U-Net com 5 ou 4 blocos de filtros convolucionais obteve resultados superiores a 0,90 de CD. Essas duas abordagens apresentaram variações nos resultados obtidos em função de qual *dataset* foi utilizado no treinamento.

Em resumo, os *datasets* com maior diversidades de características (e.g., variados estágios de crescimento da planta e falhas nas linhas de plantio, diferentes condições de luz e sombra, variadas cores de solo) são os mais indicados para o treinamento, ao produzirem

os melhores resultados e mais estáveis para outros *datasets*. A combinação de imagens utilizando uma satisfatória quantidade de imagens pode ser ainda melhor, possibilitando a rede a aprender características que são complicadas de serem reproduzidas com técnicas simples de aumento de dados.

Alguns exemplos de segmentação da Configuração 09 com o treinamento no *dataset* E(500) podem ser visualizados nas Figuras 19 e 20. É importante ressaltar que as imagens utilizadas nos exemplos (Figura 19 e nos próximos) foram selecionadas na tentativa demonstrar as principais características em análise no momento. A primeira coluna contém a imagem original, com exemplos de todos *datasets*. Na segunda coluna o *ground truth*, ou seja, a marcação realizada pelo especialista. Na terceira coluna a predição feita pela U-Net. Por fim, a quarta coluna contém a sobreposição das duas colunas anteriores, marcação × predição. Na sobreposição, as cores estão relacionados a corretude do método proposto. Branco para locais preditos corretamente como linha de plantio (Verdadeiro Positivo - VP). Preto para locais preditos corretamente como solo (Verdadeiro Negativo - VN). Vermelho para locais preditos erradamente como linha de plantio (Falso Positivo - FP) e azul para locais preditos erradamente como solo (Falso Negativo - FN).

Nos exemplos foram selecionados cenários com diferentes condições (cana-planta e cana-soca, solo mais amarelado e mais avermelhado, variadas inclinações de linhas de plantio, com a presença de falhas no plantio e estradas para o maquinário agrícola). A U-Net utilizando a Configuração 01 e a Configuração 09 não obteve bons resultados em locais com pequenas falhas na linha de cultivo e com algumas linhas de plantio que se encontram no meio do terreno. Por outro lado, obteve bons resultados para diferentes estágios de crescimento, locais com variações de iluminação e a identificação correta de solo para as estradas utilizadas pelo maquinário agrícola (locais de reorientação). Assim, pode-se afirmar que para pequenos espaços sem a cultura o método não obteve bons resultados. Entretanto, para locais mais visivelmente separados e com variados estágios de crescimento, conseguiu identificar corretamente as linhas de plantio e o solo.

5.3 Estudo de caso dos IVs na U-Net

Na literatura correlata foram encontrados trabalhos utilizando Índices de Vegetação de modo a ajudar métodos de segmentação em variados cenários da AP. Na identificação das linhas de plantio, alguns IVs, como o ExG e NDVI, foram utilizados em conjunto com métodos clássicos e redes neurais (SOUZA et al., 2017; GARCÍA-SANTILLÁN et al., 2017; PEREIRA JÚNIOR et al., 2020a; PEREIRA JÚNIOR et al., 2020b). Posto isso, a utilização de IVs em conjunto com redes neurais tornou-se uma hipótese interessante a ser analisada. A partir dos bons resultados anteriores com a U-Net, ela foi escolhida como rede neural a ser analisada com 10 dos mais utilizados IVs (Tabela 1).

O código da U-Net foi alterado para aceitar como entrada imagens com número de

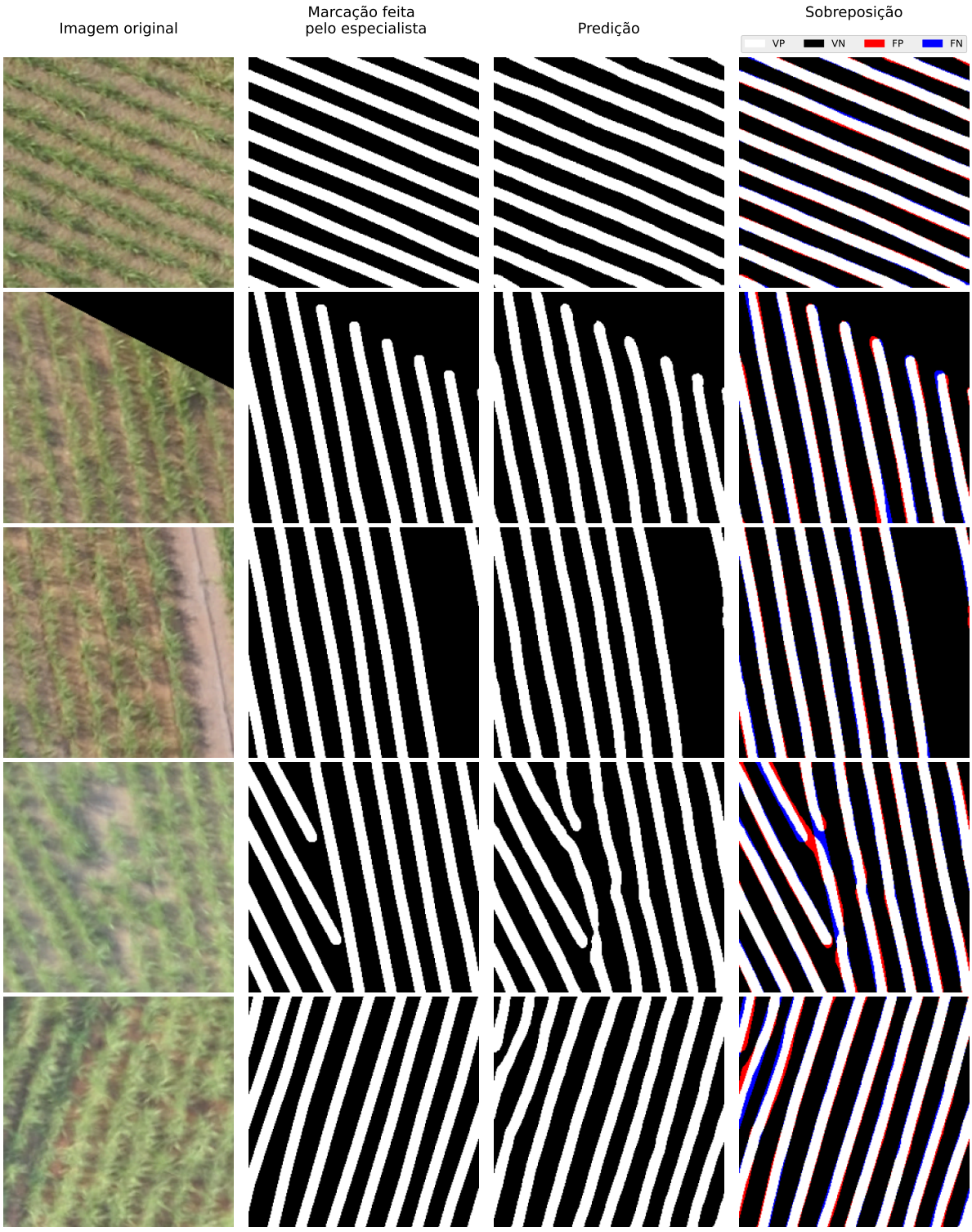


Figura 19 – Exemplos de segmentação da U-Net com a configuração 09 – parte 1: na primeira coluna a imagem original, na segunda a marcação realizada pelo especialista, na terceira a predição feita pelo método na Configuração 09 e na quarta a sobreposição das duas colunas anteriores (marcação × predição). Na sobreposição as cores denotam os resultados, branco para Verdadeiro Positivo, preto para Verdadeiro Negativo, vermelho para Falso Positivo e azul para Falso Negativo.

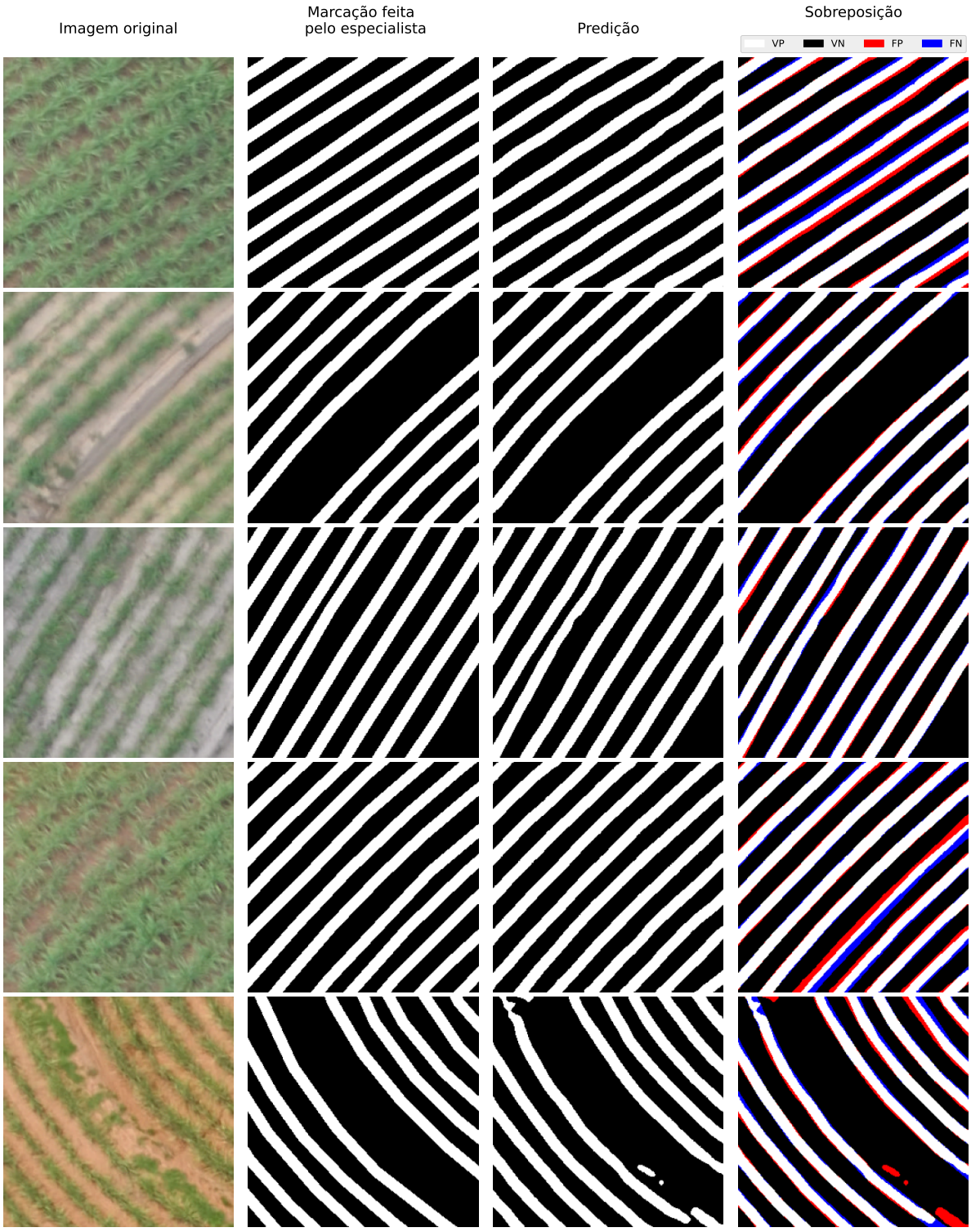


Figura 20 – Exemplos de segmentação pela U-Net com a configuração 09 – parte 2: na primeira coluna a imagem original, na segunda a marcação realizada pelo especialista, na terceira a predição feita pelo método na Configuração 09 e na quarta a sobreposição das duas colunas anteriores (marcação × predição). Na sobreposição as cores denotam os resultados, branco para Verdadeiro Positivo, preto para Verdadeiro Negativo, vermelho para Falso Positivo e azul para Falso Negativo.

camadas diferente de 3 camadas ($256 \times 256 \times 3$). A quarta camada é formada por um dos IVs selecionados em cada combinação do experimento. Cada IV é calculado e tem como resultado uma camada do tamanho da imagem ($256 \times 256 \times 1$), que depois é verificado a existência de valores *NaN* e *Inf*.

O *NaN* (do inglês *Not a Number*, não é um número) representa um valor numérico indefinido, no caso o resultado de 0/0. O *Inf* (do inglês *Infinity*, Infinito), tem função similar, podendo ser negativo ($x/0 = -Inf$, para $x < 0$) ou positivo ($x/0 = Inf$, para $x > 0$). Todos os valores de *NaN* e $\pm Inf$ são alterados para 0. A remoção dos valores *NaN* e $\pm Inf$ e a normalização tem o objetivo de facilitar e otimizar o aprendizado da U-Net. Após normalizado, os valores do IV são normalizados entre 0 e 1, através da Equação 14. Na sequência o IV normalizado é concatenado na imagem de entrada (i.e., RGB & IV), também normalizada entre 0 e 1 como no estudo apresentado na seção 5.1 (através da Equação 13).

$$valor\_normalizado = (valor - min\_valor)/(max\_valor - min\_valor) \tag{14}$$

Nos experimentos anteriores foi possível carregar e processar todas as imagens de cada *dataset* de uma única vez na memória da placa de vídeo. Por outro lado, com a nova camada adicional (e porventura outras) foi necessário processar as imagens por etapas, utilizando uma função "geradora", que divide as imagens em lotes específicos e efetua as operações necessárias (e.g., cálculo do IV) e depois os "envia" para a rede neural.

A U-Net foi treinada no *dataset* E(500) na Configuração 01 ([16, 32, 64, 128, 256]) e na Configuração 09 ([16, 16, 16, 16, 16]), devido aos seus bons resultados no estudo de caso anterior. A rede foi treinada como os mesmo hiperparâmetros utilizados anteriormente (e.g., 50 épocas, otimizador Adam com taxa de aprendizado de 0,001, 80% das imagens para treinamento e 20% para teste, e tamanho de lote de 8). Na Tabela 15 podem ser visualizados os resultados para a Configuração 01 e na Tabela 16 para a Configuração 09. A linha com "00_none" representa que foi utilizado apenas as bandas RGB, ou seja, nenhum IV foi concatenado. Por outro lado, nas outras linhas um dos IVs foi utilizado em conjunto com as bandas RGB.

Os resultados foram um pouco diferentes dos presentes na Tabela 14, com uma variação de cerca de 0,03 na média do CD, provavelmente devido as necessárias modificações feitas no código (para adicionar suporte aos IVs) e em função dos valores iniciais aleatórios dos pesos da rede. Apesar dos melhores resultados na Tabela 15 e na Tabela 16 terem sido obtidos com uso de algum IV, eles são muito próximos entre si.

A pequena variação nos resultados provavelmente foi pelas bandas RGB já forneceram todas as informações necessárias para o aprendizado da rede neural. Em outras palavras, os IVs não adicionaram diretamente nenhuma nova informação para U-Net que já não estava presente nas bandas RGB e possivelmente já haviam sido apreendidas pelos filtros

Tabela 15 – Resultados do CD médio obtidos na U-Net na Configuração 01 ([16, 32, 64, 128, 256]) com as bandas RGB e um IV.

IV	***Dataset***			
	A	B	C	D
00_none	0,9420	0,9332	0,9211	0,8872
01_VARI	0,9552	0,9361	0,9294	0,8923
02_ExG	0,9538	**0,9408**	0,9299	0,8964
03_ExR	0,9483	0,9388	0,9282	0,8954
04_ExB	0,9521	0,9406	0,9299	**0,8965**
05_ExGR	0,9507	0,9380	0,9305	0,8913
06_GRVI	**0,9568**	0,9389	**0,9329**	0,8963
07_MGRVI	0,9541	0,9335	0,9286	0,8897
08_GLI	0,9456	0,9351	0,9244	0,8848
09_RGBVI	0,9414	0,9355	0,9211	0,8948
10_IKAW	0,9469	0,9345	0,9250	0,8923

Tabela 16 – Resultados do CD médio obtidos na U-Net na Configuração 09 ([16, 16, 16, 16, 16]) com as bandas RGB e um IV.

IV	***Dataset***			
	A	B	C	D
00_none	0,9289	0,9266	0,9145	0,8754
01_VARI	0,9279	0,9304	0,9164	0,8817
02_ExG	0,9317	0,9306	0,9175	0,8859
03_ExR	**0,9325**	**0,9333**	**0,9188**	0,8852
04_ExB	0,9301	0,9290	0,9179	**0,8865**
05_ExGR	0,9319	0,9297	0,9165	0,8773
06_GRVI	0,9316	0,9312	0,9162	0,8815
07_MGRVI	0,9300	0,9274	0,9157	0,8810
08_GLI	0,9280	0,9266	0,9152	0,8765
09_RGBVI	0,9288	0,9208	0,9155	0,8784
10_IKAW	0,9295	0,9297	0,9157	0,8790

convolucionais. Ao se considerar que os IVs são (apenas) combinações algébricas de várias bandas espectrais, faz sentido não ter nenhuma real melhoria no desempenho ao se utilizar eles em conjunto com RGB. Essas combinações algébricas dos IVs provavelmente têm algumas operações similares nos vários filtros e nos pesos apreendidos pela rede.

Após a avaliação da utilização individual dos IVs, surgiu a hipótese sobre utilizar mais de um IV por vez. Como esperado, utilizar mais de um IV em conjunto com as bandas RGB teve resultado similar ao utilizar apenas um. Com resultados muito próximos entre si, que podem ser visualizado na Tabela 17 para a Configuração 01 e na Tabela 18 para a Configuração 09. A linha com "00_none" representa o uso apenas das bandas RGB (resultado obtido anteriormente). Já a linha com "01 até 05" também foi utilizado os IVs 01 até o 05 (i.e., 01_VARI, 02_ExG, 03_ExR, 04_ExB e 05_ExGR), com 8 camadas na imagem de entrada. Na linha com "06 até 10" foi utilizado os IVs de 06 até 10 (i.e.,

06_GRVI, 07_MGRVI, 08_GLI, 09_RGBVI e 10_IKAW), enquanto a linha com "01 até 10" foi utilizado todos IVs (Tabela 1), com 13 camadas na imagem de entrada. Devido ao alto número de possíveis combinações dos IVs e ao tempo necessário para executar os treinamentos da rede em cada combinação, os experimentos foram limitados a essas 3 opções com maior probabilidade de influenciar nos resultados.

Tabela 17 – Resultados do CD médio obtidos na U-Net na Configuração 01 ([16, 32, 64, 128, 256]) com as bandas RGB e vários IVs.

IVs	***Dataset***			
	A	B	C	D
00_none	0,9420	0,9332	0,9211	0,8872
01 até 05	0,9544	0,9384	0,9323	0,8972
06 até 10	**0,9587**	**0,9395**	**0,9353**	**0,8996**
01 até 10	0,9507	0,9363	0,9290	0,8949

Tabela 18 – Resultados do CD médio obtidos na U-Net com a Configuração 09 ([16, 16, 16, 16, 16]) com as bandas RGB e vários IVs.

IVs	***Dataset***			
	A	B	C	D
00_none	0,9289	0,9266	0,9145	0,8754
01 até 05	0,9302	0,9253	0,9136	0,8751
06 até 10	**0,9321**	**0,9308**	**0,9175**	**0,8818**
01 até 10	0,9305	0,9305	0,9158	0,8779

Com base nos resultados obtidos utilizando os IVs como dados complementares, não se pode afirmar muito devido à proximidade deles. Entretanto, a avaliação dessa hipótese foi importante dado a utilização dos IVs em trabalhos de detecção de linhas de plantio. Considerando que os IVs não adicionaram muito para o aprendizado da U-Net, outra hipótese foi avaliada, utilizando apenas os IVs, sem as bandas RGB.

Nesse cenário, os resultados foram um pouco diferentes, como pode ser observado na Tabela 19 para a Configuração 01 e na Tabela 20 para a Configuração 09. Apesar dos resultados serem ligeiramente piores do que os utilizando as bandas RGB, são bem próximos, podendo ser uma opção viável para determinados cenários (e.g., locais com limitado poder computacional). Os IVs de 01 até 09 tiveram resultados próximos, contudo o 10_IKAW não teve resultados tão bons, assim não é recomendado para ser utilizado como única entrada para a U-Net. O 04_ExB não propiciou uma boa generalização das características apreendidas entre os *datasets*, tendo um desempenho inferior nos *datasets* B e D.

Os outros IVs tiveram resultados similares, ficando estaticamente muito próximos. Entretanto, o 05_ExGR (05_ExGR = 02_ExG − 03_ExR) tem a desvantagem que é necessário calcular os outros dois IVs para depois calcular seu valor de fato. Portanto,

Tabela 19 – Resultados do CD médio obtidos na U-Net com a Configuração 01 utilizando apenas os IVs, um por vez e sem utilizar as bandas RGB.

IV	***Dataset***			
	A	B	C	D
01_VARI	0,9282	0,9253	0,9116	0,8781
02_ExG	0,9310	0,9181	**0,9179**	0,8762
03_ExR	0,9300	0,9154	0,9085	0,8793
04_ExB	0,9208	0,8760	0,9034	0,8438
05_ExGR	**0,9345**	**0,9287**	0,9170	**0,8825**
06_GRVI	0,9178	0,9263	0,9079	0,8769
07_MGRVI	0,9307	0,9260	0,9150	0,8794
08_GLI	0,9325	0,9265	0,9150	0,8749
09_RGBVI	0,9326	0,9231	0,9155	0,8775
10_IKAW	0,8857	0,8961	0,8504	0,8387

Tabela 20 – Resultados do CD médio obtidos na U-Net com a Configuração 09 com apenas IVs, um por vez e sem utilizar as bandas RGB.

IV	***Dataset***			
	A	B	C	D
01_VARI	0,9225	0,9243	0,9101	0,8735
02_ExG	0,9247	0,9242	0,9126	0,8731
03_ExR	0,9263	0,9240	0,9112	**0,8771**
04_ExB	0,9108	0,8774	0,8984	0,8287
05_ExGR	**0,9270**	**0,9234**	0,9120	0,8723
06_GRVI	0,9253	0,9253	0,9111	0,8751
07_MGRVI	0,9248	0,9255	0,9135	0,8761
08_GLI	0,9257	0,9246	**0,9139**	0,8765
09_RGBVI	0,9240	0,9222	0,9127	0,8710
10_IKAW	0,8560	0,8973	0,8496	0,8309

não é uma boa escolha para cenários com poder computacional limitado, como sistemas embarcados ou computadores acoplados no maquinário agrícola.

O 01_VARI, apesar dos bons resultados, em função do seu cálculo (VARI = $(g - r)/(g + r - b)$) apresenta muitos valores que precisam ser "corrigidos" (i.e., alterado para valores numéricos em vez de *NaN* e *Inf*). Assim, é interessante evitar IVs com a operação de divisão no seu cálculo, devido à necessária verificação dos valores "não numéricos" após o cálculo. Também é interessante evitar IVs com a operação de exponenciação, devido ao seu custo computacional. Dessa forma, o 02_ExG é um dos mais interessante a ser utilizado, já que seu cálculo (ExG = $2*g-r-b$) é um dos mais simples quando comparado aos outros IVs avaliados e não tem nenhuma operação de divisão ou de exponenciação.

Também foi avaliado a utilização de mais de um IVs sem as bandas RGB. Os resultados da Configuração 01 podem ser visualizados na Tabela 21 e da Configuração 09 na Tabela 22. Como os resultados são muito similares aos utilizando apenas um IV, é de pouco ajuda utilizar mais de um IV como única entrada para a U-Net.

Tabela 21 – Resultados do CD médio obtidos na U-Net com a Configuração 01 ([16, 32, 64, 128, 256]) sem RGB e vários IVs.

IVs	***Dataset***			
	A	B	C	D
00_none	0,9420	0,9332	0,9211	0,8872
01 to 05	0,9378	0,9272	0,9162	0,8808
06 to 10	0,9356	0,9197	0,9192	0,8827
01 to 10	0,9276	0,9241	0,9139	0,8831

Tabela 22 – Resultados do CD médio obtidos na U-Net com a Configuração 09 ([16, 16, 16, 16, 16]) sem RGB e vários IVs.

IVs	***Dataset***			
	A	B	C	D
00_none	0,9289	0,9266	0,9145	0,8754
01 to 05	0,9290	0,9230	0,9130	0,8760
06 to 10	0,9297	0,9208	0,9129	0,8759
01 to 10	0,9291	0,9195	0,9091	0,8738

A utilização de um IV em vez das bandas RGB pode ser interessante para locais com limitado poder computacional e/ou largura de banda limitada. Além de ser uma alternativa para reduzir os custos com os equipamentos necessários para a execução de todo o processo. É necessário calcular o IV para cada imagem, entretanto esse processo reduz de três para uma camada a ser processada pela rede neural. Essa redução de 1/3, também ocorre na utilização na banda de conexão (caso as imagens forem processadas em outro computador/dispositivo), além de um possível menor uso de GPU, CPU e RAM no processamento final.

Para título de comparação, o treinamento de cada época na Configuração 01 (i.e., [16, 32, 64, 128, 256]) da U-Net utilizado as bandas RGB e um IV (concatenados) no *dataset* E(500) (Tabela 15) levou em média 60 segundos. Tempo similar ao se utilizar apenas as bandas RGB (Tabela 14). Por outro lado, o mesmo treinamento utilizando apenas um IV (sem as bandas RGB) (Tabela 17) levou em média 57 segundos.

É importante destacar que o treinamento da rede neural é, em sua maioria, demorado (questão de horas dependendo dos hiperparâmetros e do *dataset* utilizado), entretanto, a predição é, geralmente, bem rápida (questão de milissegundos para cada imagem). Assim, a utilização de um modelo na predição pode ser viável mesmo em locais com limitação de poder computacional. Desta forma, a utilização da Configuração 09 em conjunto de um IVs, como o ExG, pode viabilizar a sua utilização mesmo em cenários com grande limitação de poder computacional, como em sistemas embarcados acoplados no maquinário agrícola.

5.4 Estudo de caso do *dataset* do LAPIX

Com os bons resultados anteriores, torna-se importante avaliar o método em outros *datasets*. Entretanto, a disponibilidade deles é bem rara, ainda mais com as marcações de um especialista. O único encontrado foi o *dataset* L (PEREIRA JÚNIOR; WANGENHEIM, 2019) do LAPIX, que pode ser visualizado na Figura 16.

Nesse *dataset* a marcação foi realizada por um especialista de modo diferente da realizada nos *datasets* utilizados nos outros experimentos. O especialista selecionou apenas pixels contendo a cultura (como pode ser observado na Figura 21), e também é afirmado da não existência de ervas daninhas. Primeiramente, foi necessário alterar os pixels nesse mosaico para ficar conforme os utilizados anteriormente no treinamento da U-Net. Assim, foi alterado o solo (em vermelho) para preto e as linhas de plantio (em verde) para branco (como pode ser visualizado na Figura 22).

Para a realização dos experimentos, o *dataset* L foi recortado em pedaços de 256 × 256 pixels sem sobreposição. Foram descartadas as amostras com menos de 80% de informação útil (i.e., pixels com valores diferentes de zero). Após o descarte, o *dataset* L ficou com 406 imagens de um total de 962 imagens (407 totalmente pretas, 62 menores que 256 × 256, e 87 com menos de 80% de informação útil).

Inicialmente foram selecionadas as redes treinadas no *dataset* E(500) (seção 5.1, Tabela 6, linha E(500)) para teste no *dataset* L, não obtendo bons resultados, como pode ser observado na Tabela 23. Na sequência, as imagens do *dataset* L foram aplicadas na Configuração 01 e na Configuração 09 da U-Net, configurações também treinadas com o *dataset* E(500) e ambas sem a utilização de IVs (seção 5.2, Tabela 14).

Tabela 23 – Resultados do CD médio obtidos para cada rede após treinamento no *dataset* E(500) e teste no *dataset* L

Rede	**Teste**
	L
U-Net	0,6851
LinkNet	0,7210
PSPNet	0,7054

Como era esperado, a predição apresentou um resultado inferior quando comparada com os experimentos anteriores (como pode ser analisado na Tabela 24), tendo em vista que *dataset* L foi marcado pelo especialista de outra forma (i.e., apenas os pixels que representam a cultura). Com a marcação feita dessa forma, o *dataset* L tem menos linhas marcadas que o *dataset* utilizado no treinamento e também vários locais que pelo método de marcação anterior (i.e., uma linha sólida representando a linha de plantio) seriam entendidos com falhas no plantio. Assim, o CD médio foi significativamente inferior aos dos testes nos outros *datasets*.

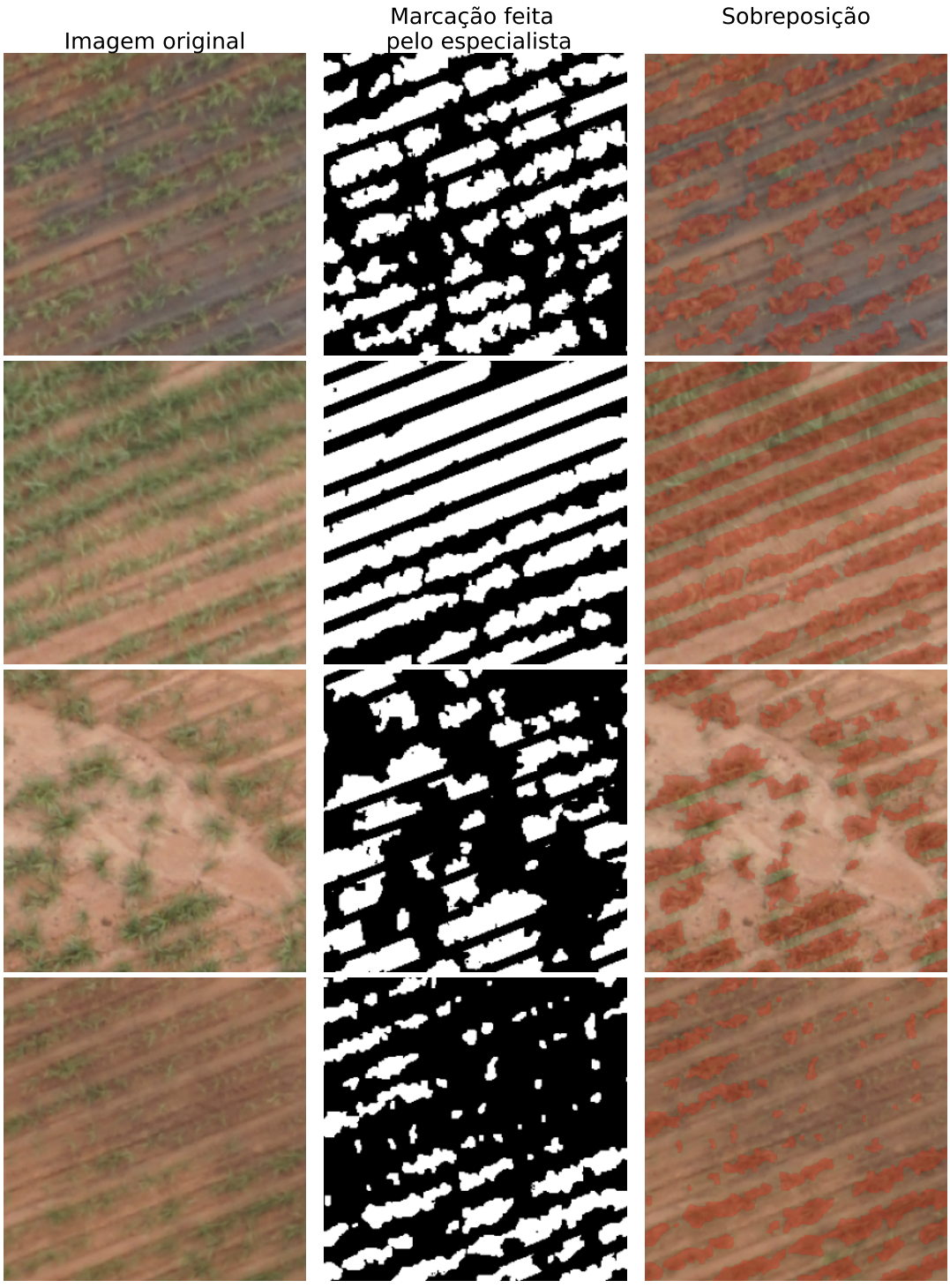


Figura 21 – Exemplos de marcações realizadas pelo especialista no mosaico Pereira Júnior e Wangenheim (2019). A primeira coluna contém a imagem original e na segunda a marcação realizada pelo especialista. Por fim, na terceira coluna, uma sobreposição das duas colunas anteriores, em vermelho a marcação do especialista.

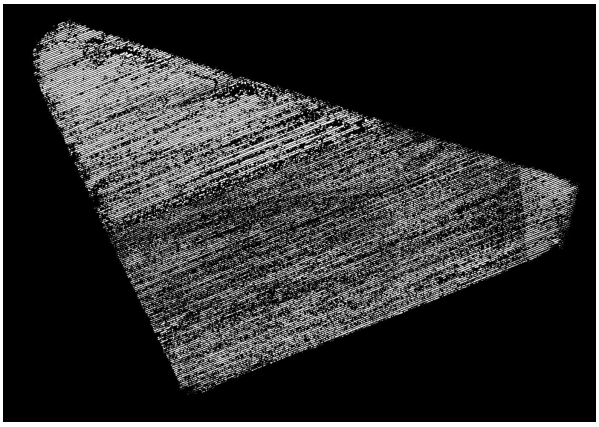


Figura 22 – Mosaico Pereira Júnior e Wangenheim (2019) após alteração dos pixels que representam o solo para preto e os pixels que representam a cultura (cana-de-açúcar) para branco.

Tabela 24 – Resultados do CD médio obtidos nos modelos treinados na U-Net com *dataset* E(500).

	A	B	C	D	L
Configuração 01	0,9420	0,9332	0,9211	0,8872	0,6760
Configuração 09	0,9289	0,9266	0,9145	0,8754	0,6682

Ao treinar no *datasets* E(500) (e consequentemente nos *datasets* A, B, C e D) a U-Net (e também as outras redes) aprendeu a segmentar a linha de plantio ligando pequenas falhas (como pode ser averiguado na Figura 23). Ao tentar fazer o mesmo no *dataset* L, isso resultou em um baixo CD médio. Um modelo treinado deste modo pode ser útil para identificar e conectar as linhas de plantio, e ser utilizado com entrada para outro processo ou método.

A partir dos resultados inferiores do teste no *dataset* L, torna-se relevante a avaliação com treinamento e teste com o *dataset* L. Assim, para o próximo teste as redes da seção 5.1 foram treinadas no *dataset* L e aplicadas nos cinco *datasets*. O treinamento utilizou a mesma configuração da seção 5.1, entretanto sem a utilização da validação cruzada *k-fold*, mas com a divisão de 90% dos dados para treinamento e 10% para teste. Os resultados para os *dataset* A, B, C e D ficaram bem inferiores, contudo os resultados para o *dataset* L melhoraram bastante, como pode ser analisado na Tabela 25.

Tabela 25 – Resultados do CD médio obtidos para cada rede após treinamento no *dataset* L e teste nos cinco *datasets*

Rede	**Teste**				
	A	B	C	D	L
U-Net	0,6663	0,7177	0,6769	0,6496	0,9072
LinkNet	0,6858	0,7244	0,6772	0,6495	0,8997
PSPNet	0,5977	0,6203	0,6255	0,6022	0,7312

Esses resultados demonstram que o método de marcação utilizado pelos especialistas,

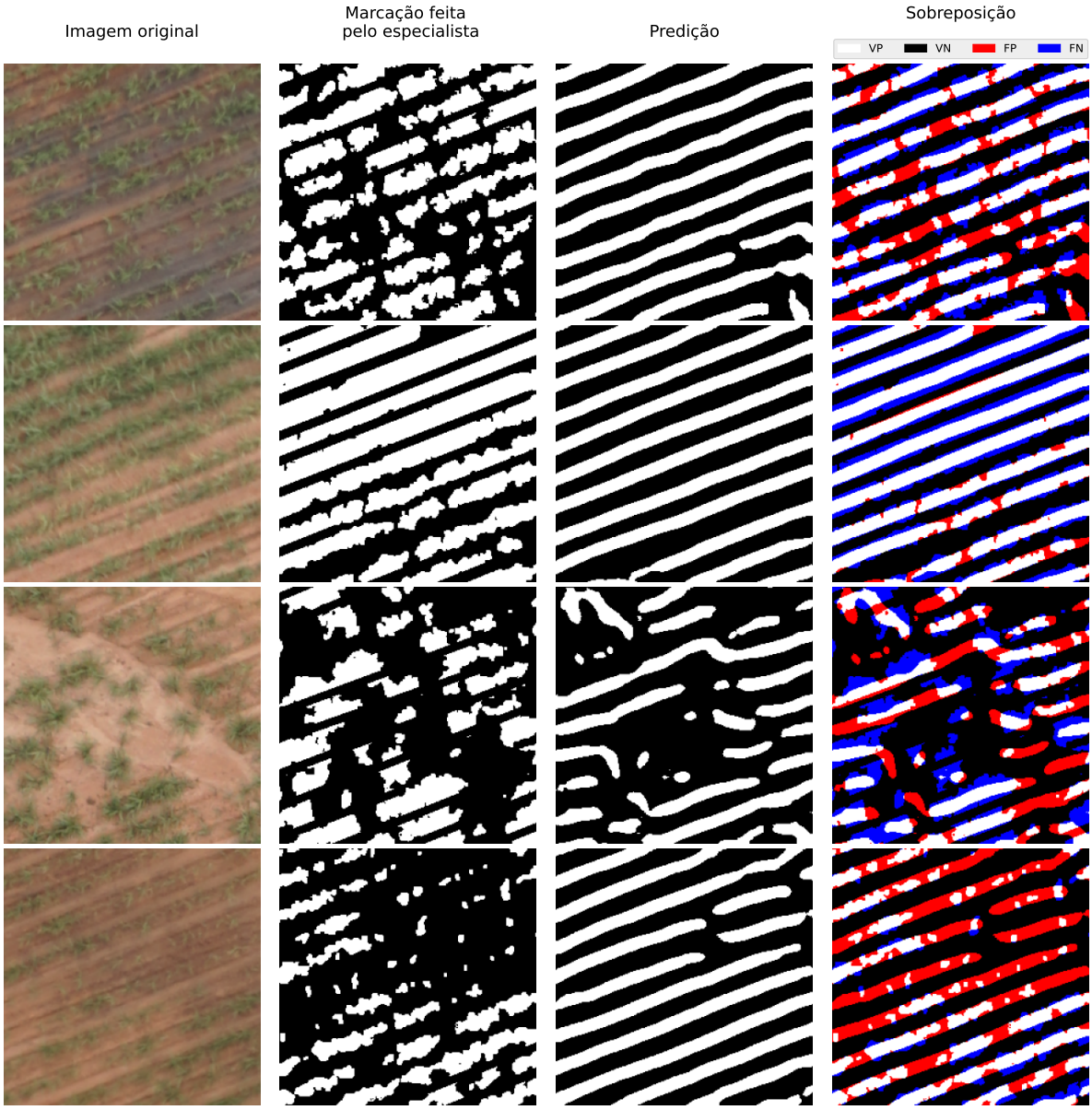


Figura 23 – Exemplos de predição do *dataset* L pelo modelo treinado com o *dataset* E(500) na U-Net com a Configuração 09. Na primeira coluna a imagem original, na segunda a marcação realizada pelo especialista, na terceira a predição feita pela U-Net com a Configuração 09 e na quarta a sobreposição das duas colunas anteriores (marcação × predição). Na sobreposição as cores denotam os resultados, branco para Verdadeiro Positivo, preto para Verdadeiro Negativo, vermelho para Falso Positivo e azul para Falso Negativo.

que por conseguinte definem o estilo de segmentação das redes neurais artificias treinadas, pode influenciar bastante no resultado. Isso sem considerar os possíveis erros de marcação cometidos pelos especialistas.

Utilizando o *dataset* L no treinamento (com os mesmo hiperparâmetros utilizados anteriormente, e.g., 50 épocas, otimizador Adam com taxa de aprendizado de 0,001, 80% das imagens para treinamento e 20% para teste, e tamanho de lote de 8) e na predição o resultado muda bastante, apresentando bons resultados para o *dataset* L, mas ruins para os outros (como pode ser analisado na Tabela 26).

Tabela 26 – Resultados do CD médio dos modelos treinados na U-Net com *dataset* L.

	A	B	C	D	L
Configuração 01	0,5530	0,6375	0,5836	0,5730	0,9179
Configuração 09	0,5196	0,6177	0,5608	0,5621	0,9044

A U-Net a partir do *Segmentation Models* (seção 5.1), obteve o CD médio de 0,9072, por outro lado, a U-Net implementada do zero com a Configuração 01 obteve 0,9179 e 0,9044 para Configuração 09. Corroborando a hipótese que essas duas configurações podem obter os melhores resultados.

A U-Net agora (e também as outras redes) busca segmentar apenas os pixels que julga ser da cultura e tendo resultado próximo da marcação do especialista (como pode ser averiguado na Figura 24). É importante notar que a U-Net com a Configuração 09, com menos de 50.000 parâmetros, foi capaz de apreender as características do *dataset* L de apenas 406 imagens, menor quantidade que o *dataset* A. Um modelo treinado desse modo pode ser útil em técnicas que realizam estimativas de produção e/ou identificação das falhas no plantio.

5.5 Estudo de caso das Operações Morfológicas

Com base nos resultados já obtidos, torna-se relevante as tentativas de melhorá-lo o resultado. Em outros trabalhos foram utilizadas técnicas como a Transformada de Hough e a Transformada de Radon (SILVA; ESCARPINATI; BACKES, 2021), contudo sem real melhoria.

Dentre as técnicas de PDI, as operações morfológicas não têm grande custo computacional e podem melhorar o resultado, assim foram escolhidas para avaliação e tentativa de melhoria do resultado. Dois elementos estruturantes de tamanho 3×3 foram escolhidos para serem aplicados nas operações morfológicas, Todos_1, com todos os valores como 1 e Cruz, com os valores de 1 formando uma cruz, como poder ser visualizado na Tabela 27. Dentre as operações morfológicas, algumas das mais utilizadas foram selecionadas: a dilatação, a erosão, a abertura e o fechamento.

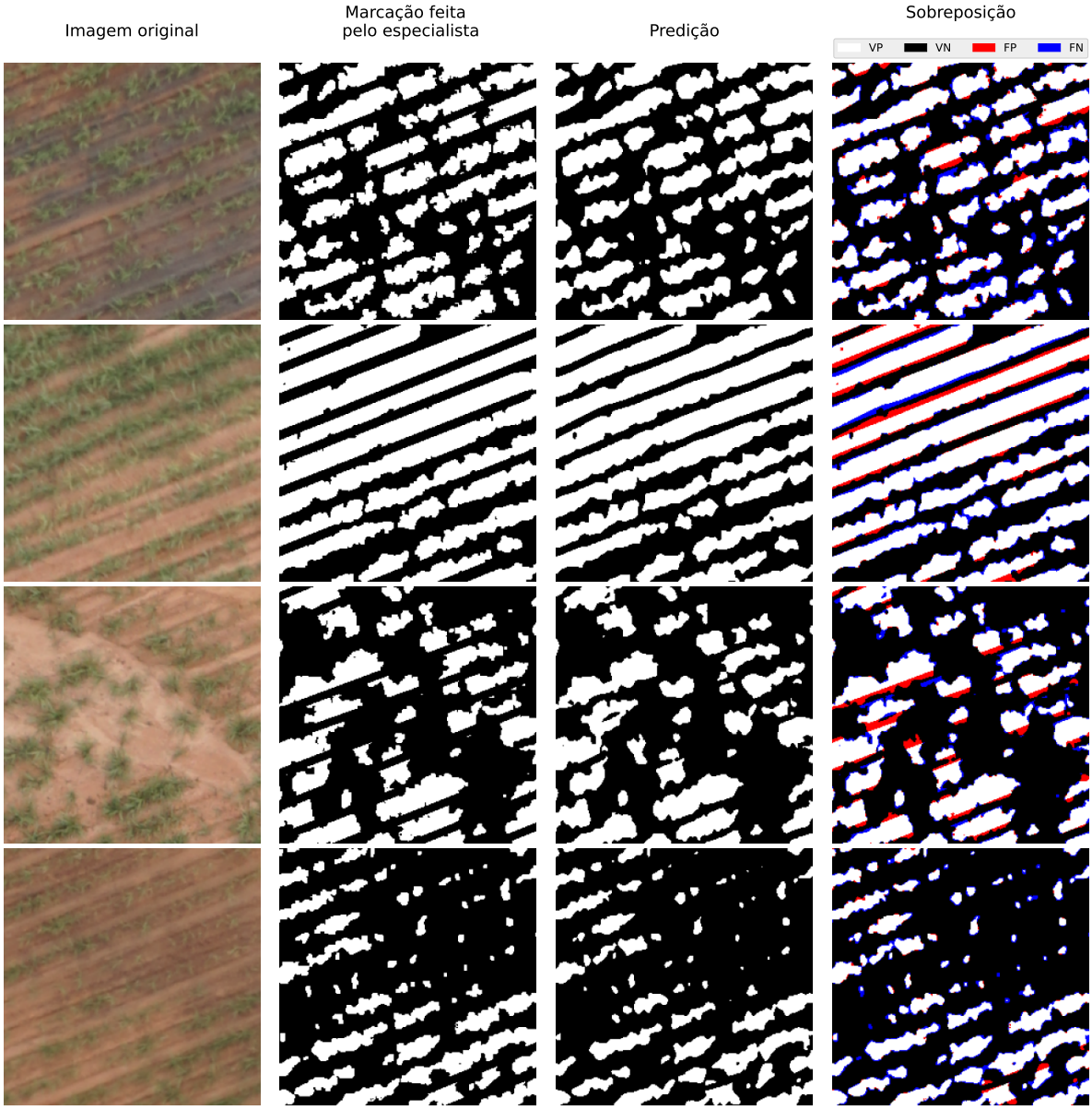


Figura 24 – Exemplos de predição do *dataset* L pelo modelo treinado com o *dataset* L na U-Net com a Configuração 09. Na primeira coluna a imagem original, na segunda a marcação realizada pelo especialista, na terceira a predição feita pela U-Net com a Configuração 09 e na quarta a sobreposição das duas colunas anteriores (marcação × predição). Na sobreposição as cores denotam os resultados, branco para Verdadeiro Positivo, preto para Verdadeiro Negativo, vermelho para Falso Positivo e azul para Falso Negativo.

Tabela 27 – Elementos estruturantes utilizado nas operações morfológicas.

Todos_1			**Cruz**		
1	1	1	0	1	0
1	1	1	1	1	1
1	1	1	0	1	0

Para comparação foram escolhidos os resultados da Configuração 01 na U-Net com treinamento no *dataset* E(500), utilizando as bandas RGB e nenhum IVs, (a linha com "00_none" da Tabela 15). Cada combinação de operação morfológica e elemento estruturante foram aplicadas nesses resultados (em todas as imagens) e depois foi calculado o CD médio. Os resultados podem ser visualizados na Tabela 28.

Tabela 28 – Resultados do CD médio após as operações morfológicas na segmentação da U-Net na Configuração 01 treinada com o *dataset* E(500).

	IV	A	B	C	D
Configuração 01	00_none	0,9420	0,9332	0,9211	0,8872

Operação	Elemento	A	B	C	D
Erosão	Cruz	0,8969	0,8975	0,8768	0,8535
	Todos_1	0,8798	0,8625	0,8571	0,8320
Dilatação	Cruz	0,9193	0,9159	0,9010	0,8774
	Todos_1	0,9099	0,8951	0,8902	0,8668
Abertura	Cruz	0,9320	0,9286	0,9146	0,8846
	Todos_1	0,9419	0,9331	0,9210	0,8870
Fechamento	Cruz	0,9320	0,9286	0,9146	0,8846
	Todos_1	0,9419	0,9331	0,9211	0,8872

A aplicação das operações morfológicas em todas as imagens não resultaram em nenhuma melhoria no resultado, mas sim em uma piora no CD médio. Os elementos estruturantes também não propiciaram real diferença entre si nos resultados.

Tabela 29 – Resultados do CD médio após as operações morfológicas na segmentação da U-Net na Configuração 09 treinada com o *dataset* L.

	IV	L
Configuração 09	00_none	0,9044

Operação	Elemento	L
Erosão	Cruz	0,8319
	Todos_1	0,7999
Dilatação	Cruz	0,9062
	Todos_1	0,8927
Abertura	Cruz	0,9020
	Todos_1	0,9005
Fechamento	Cruz	0,9020
	Todos_1	0,9052

Na Figura 25 podem ser visualizados alguns exemplos de aplicação das operações morfológicas utilizando o elemento estruturante Todos_1 em predições da Configuração 01 da U-Net, treinada e testada no *dataset* E(500). Por outro lado, na Figura 26 podem ser visualizados alguns exemplos de aplicação das operações morfológicas utilizando o

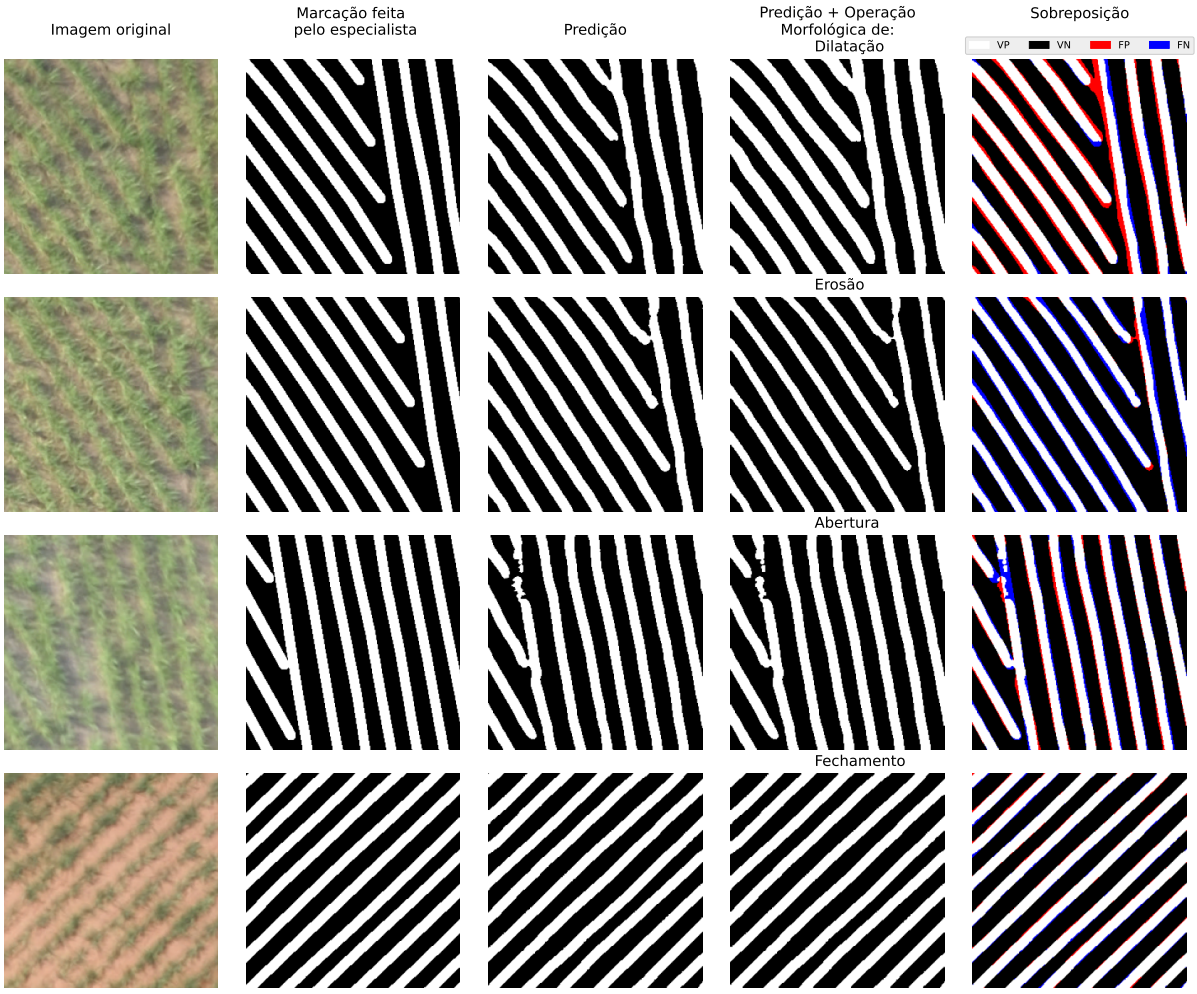


Figura 25 – Exemplos de aplicação das operações morfológicas, utilizando o elemento estruturante Todos_1, em predições da U-Net com a Configuração 01, treinada no *dataset* E(500) e feita a predição do mesmo. Na primeira coluna a imagem original, na segunda a marcação realizada pelo especialista, na terceira a predição feita pela U-Net com a Configuração 01, na quarta o resultado da operação morfológica (nome da operação sobre a imagem) e na quinta a sobreposição da marcação × predição após operação morfológica. Na sobreposição as cores denotam os resultados, branco para Verdadeiro Positivo (VP), preto para Verdadeiro Negativo (VN), vermelho para Falso Positivo (FP) e azul para Falso Negativo (FN).

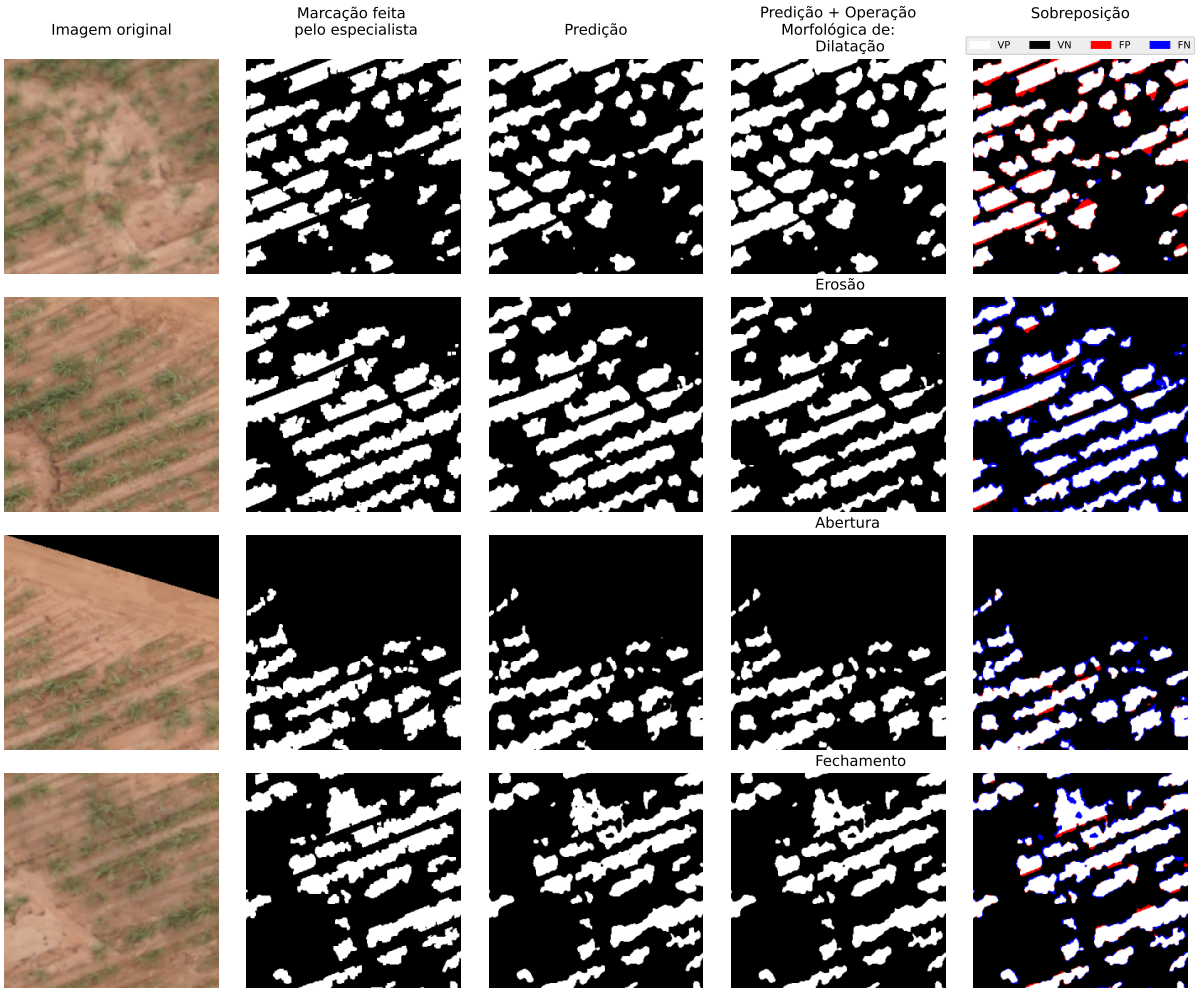


Figura 26 – Exemplos de aplicação das operações morfológicas, utilizando o elemento estruturante Cruz, em predições da U-Net com a Configuração 09, treinada no *dataset* L e feita a predição do mesmo. Na primeira coluna a imagem original, na segunda a marcação realizada pelo especialista, na terceira a predição feita pela U-Net na Configuração 09, na quarta o resultado da operação morfológica (nome da operação sobre a imagem) e na quinta a sobreposição da marcação × predição após operação morfológica. Na sobreposição as cores denotam os resultados, branco para Verdadeiro Positivo (VP), preto para Verdadeiro Negativo (VN), vermelho para Falso Positivo (FP) e azul para Falso Negativo (FN).

elemento estruturante Cruz em predições da Configuração 09 da U-Net, treinada e testada no *dataset* L.

Os resultados (imagens) após a aplicação das operações morfológicas utilizando o elemento estruturante Todos_1 para os com o elemento estruturante Cruz tiveram pouca diferença em uma análise visual, assim como no novo CD médio obtido. Para o *dataset* L essa diferença foi um pouco mais visível, obtendo pequenas melhorias na identificação dos pixels refentes a cultura, contudo não foi uma melhoria significativa. Mesmo no melhor dos casos, utilizando a operação morfológica Dilatação com o elemento estruturante Cruz, o no novo CD médio (0,9062) foi muito similar ao obtido (0,9044) antes da aplicação da operação.

Capítulo 6

Conclusão

O principal objetivo deste projeto foi analisar alguns dos principais métodos utilizados para segmentar imagens capturadas por VANTs, com foco em imagens de plantações de cana-de-açúcar e a partir deles, propor melhorias na detecção das linhas de plantio. Inicialmente foi identificado na literatura correlata a utilização de métodos tradicionais de aprendizado de máquina (SVM e KNN), para segmentar as imagens. Na sequência, foi identificado a utilização de outros métodos (AG, método Otsu, TH e TR) e *softwares* comerciais (*Inforow*), e em trabalhos mais recentes, a utilização de CNNs. Alguns desses trabalhos utilizaram IVs em conjunto com o método proposto.

A partir dos trabalhos relacionados encontrados e de seus bons resultados obtidos utilizando as CNNs, elas foram escolhidas para uma análise mais aprofundada. Dentre as CNNs, foram selecionadas três (U-Net, PSPNet e LinkNet) das mais utilizadas para segmentação. Nas configurações avaliadas e utilizando *datasets* de imagens reais de plantações de cana-de-açúcar de variados estágios de crescimento, a U-Net obteve os melhores resultados (Seção 5.1). Também foi criado um novo *dataset* para avaliar e confirmar a hipótese de que um *dataset* mais diverso pode resultar em resultados melhores.

Após esses resultados, mais experimentos foram desenvolvidos para avaliar quais partes da U-Net (número de blocos e filtros convolucionais) influenciam nos resultados. Das configurações analisadas (Seção 5.2), a Configuração 01 (i.e., [16, 32, 64, 128, 256]) e a Configuração 09 ([16, 16, 16, 16, 16]) obtiveram os melhores resultados.

Com base em trabalhos relacionados utilizando os IVs, foi avaliado a utilização deles no treinamento da U-Net (Seção 5.3) nas duas melhores configurações obtidas experimento anterior. A utilização dos IVs como dados de entrada complementares não resultou em significativa melhoria nos resultados. Por outro lado, a utilização de um IV como dados de entrada exclusivos pode ser uma alternativa viável para cenários com, por exemplo, limitado poder computacional.

Na sequência, para validar o método proposto, foram realizados experimentos no *dataset* L. Como a marcação nesse *dataset* foi realizada de modo diferente (marcando apenas os pixels da cultura), os modelos treinados nos outros *datasets* não tiveram bons resulta-

dos ao segmentar esse *dataset*. Entretanto, ao se realizar o treinamento com este *dataset*, os resultados ficaram bem melhores. Corroborando assim a hipótese de que os *datasets* utilizados no treinamento podem influenciar muito no resultado (segmentação) e nas características que o modelo irá analisar nas futuras imagens de entrada.

Por fim, foi realizado uma tentativa de otimizar o resultado obtido utilizando operações morfológicas. Das operações morfológicas, foram selecionadas 4 das mais utilizadas e simples (erosão, dilatação, abertura e fechamento) com duas opções de elementos estruturante (Cruz e Todos_1). Contudo, a aplicação das operações morfológicas não resultou em melhoria no resultado.

Tendo em vista os vários experimentos realizados, os objetivos estabelecidos foram alcançados com sucesso, e os resultados obtidos destacam a eficácia dos métodos de aprendizado profundo na segmentação automática de imagens capturadas por VANTs em plantações de cana-de-açúcar. Uma abordagem similar provavelmente terá bons resultados em outras culturas (como milho e café).

Abaixo tem-se um resumo das recomendações obtidas a partir dos experimentos deste projeto:

- ❏ para o treinamento da rede neural é importante construir um *dataset* com imagens de variados estágios de crescimento (cana-planta e cana-soca) e de plantações diversas. Uma alternativa é utilizar *datasets* específicos (com os estágios específicos a serem futuramente analisados/segmentados) no treinamento, gerando assim modelos com foco em certos cenários/estágios. Entretanto, essa abordagem adiciona uma análise visual das imagens para escolha de qual modelo utilizar.

- ❏ a U-Net apresentou bons resultados na segmentação das imagens, com duas configurações em destaque. A Configuração 01 ([16, 32, 64, 128, 256]) é recomendada para cenários com melhor poder computacional. Em contrapartida, a Configuração 09 ([16, 16, 16, 16, 16]) é recomendada para cenários com poder computacional mais restrito ou limitado e sem degradar muito a segmentação.

- ❏ técnicas de aumento de dados não foram muito efetivas no cenário analisado, ou pelo menos com as redes neurais analisadas.

- ❏ o uso de apenas uma camada com um IV, como o ExG, pode ter resultado similar ao utilizar as três bandas RGB, podendo gerar assim uma economia de recursos computacionais (e.g., em processador e largura de banda).

Como restrições e limitações têm-se:

- ❏ três redes neurais foram testadas com certos hiperparâmetros/configurações. Outras redes podem obter um resultado ainda melhor ou uma nova configuração específica das redes analisadas.

- ❏ os IVs foram avaliados apenas na U-Net, assim é interessante avaliá-los em outras redes neurais e verificar se obtêm resultados similares. Também é importante avaliar outros IVs.

- ❏ devido à rara disponibilidade de *datasets* de imagens de cana-de-açúcar, os experimentos foram realizados em apenas seis *datasets*. Portanto, mais testes em outros *datasets* (com marcações de especialistas) são importantes.

- ❏ a utilização de imagens com outras bandas além das RGB (como o infravermelho próximo) e outros IVs que fazem uso dessas bandas (como o NDVI) podem propiciar resultados ainda melhores, apesar do maior custo em capturar imagens com essas bandas. Hipótese ainda a ser explorada.

Devido à raridade de *datasets* disponíveis de imagens de plantações de cana-de-açúcar capturadas por VANTs e com marcação de especialistas, não foi possível realizar uma comparação aprofundada com trabalho relacionados. Entretanto, os métodos propostos foram aplicados em alguns *datasets* e obtiveram bons resultados para a maioria deles (CD médio de 0,90 ou mais). Dentre os experimentos, a Configuração 01 e a principalmente a Configuração 09 da U-Net implementada do zero obtiveram destaque pela quantidade reduzida de parâmetros, e mesmo assim obtendo bons resultados.

6.1 Principais Contribuições

- ❏ Demonstrar a capacidade das CNNs para segmentar imagens capturadas por VANTs de plantações de cana-de-açúcar, avaliando a influência da utilização de métodos simples de aumento de dados, *transfer learning* e *fine-tuning* nos resultados com experimentos em *datasets*.

- ❏ Análise aprofundada da U-Net e de suas principais partes, com recomendações de configurações para determinados cenários.

- ❏ Avaliação da possibilidade de utilização dos IVs no treinamento da U-Net e da utilização de algumas das principais operações morfológicas para otimizar a segmentação.

6.2 Trabalhos Futuros

Os resultados deste projeto demonstram um bom desempenho obtido pela abordagem proposta e motivam novas linhas de investigação, tais como: avaliação de *datasets* de diferentes culturas além da cana-de-açúcar; explorar o uso de imagens com outras bandas, como infravermelho, e outros IVs; analisar o poder computacional necessário para

implantar o método proposto em cenários reais e avaliar a provável redução ao se utilizar a Configuração 09 em vez da Configuração 01, e apenas um IVs como dados de entrada para rede em vez das bandas RGB; comparar o método proposto com métodos comercias presentes no mercado, como o *Inforow* (INFOROW, 2021); estudar e avaliar outras redes neurais artificiais utilizadas na segmentação semântica para segmentar as linhas de plantio; utiliza alguma técnica de otimização de hiperparâmetros (como *Grid Search* - Busca em Grade) para encontrar os melhores hiperparâmetros para as novas redes avaliadas.

6.3 Contribuições em Produção Bibliográfica

Os seguintes artigos estão fortemente ligados com a pesquisa deste projeto:

- RIBEIRO, J. B., SILVA, R. R. d., DIAS JÚNIOR, J. D., ESCARPINATI, M. C., BACKES, A. R.. Automated detection of sugarcane crop lines from UAV images using deep learning. **Information Processing in Agriculture**, 2023. ISSN 2214-3173. Disponível em: <https://doi.org/10.1016/j.inpa.2023.04.001>

Submetido e em fase de análise:

- RIBEIRO, J. B., BACKES, A. R.. Adjusting convolution blocks of U-Net to improve sugarcane crop line segmentation. **Submetido ao Soft Computing**.

O estudo de conceitos sobre PDI e CNNs permitiu a contribuição com outros problemas relacionados à visão computacional, assim permitindo auxiliar no artigo:

- DIAS JÚNIOR, J. D.; RIBEIRO, J. B.; BACKES, A. R. Assessing the impact of JPEG compression on the semantic segmentation of agricultural images. **Signal, Image and Video Processing**, 2023. ISSN 1863-1711. Disponível em: <https://doi.org/10.1007/s11760-023-02697-7>.

Referências

AERO Engenharia. **Identificação de falhas em linha de plantio – A aplicação dos drones no setor sucroenergético**. 2017. Disponível em: <https://aeroengenharia.com/falhas-em-linha-de-plantio/>. Acesso em: 19 de março de 2023.

ALAZAWEE, W. S.; ABDEL-QADER, I.; ABDEL-QADER, J. Using morphological operations — Erosion based algorithm for edge detection. In: **2015 IEEE International Conference on Electro/Information Technology (EIT)**. IEEE, 2015. p. 521–525. ISSN 2154-0373. Disponível em: <https://doi.org/10.1109/EIT.2015.7293391>.

BAH, M. D.; HAFIANE, A.; CANALS, R. CRowNet: Deep Network for Crop Row Detection in UAV Images. **IEEE Access**, v. 8, p. 5189–5200, 2020. ISSN 2169-3536. Disponível em: <https://dx.doi.org/10.1109/ACCESS.2019.2960873>.

BARBOSA JÚNIOR, M. R. **Mapeamento de falhas em cana-de-açúcar por imagens de veículo aéreo não tripulado**. Dissertação (Mestrado) — Universidade Estadual Paulista (UNESP), 2021. Disponível em: <https://doi.org/11449/202810>.

BARBOSA JÚNIOR, M. R. et al. The Time of Day Is Key to Discriminate Cultivars of Sugarcane upon Imagery Data from Unmanned Aerial Vehicle. **Drones**, MDPI AG, v. 6, n. 5, Apr. 2022. ISSN 2504-446X. Disponível em: <http://dx.doi.org/10.3390/drones6050112>.

_____. Mapping Gaps in Sugarcane by UAV RGB Imagery: The Lower and Earlier the Flight, the More Accurate. **Agronomy**, Multidisciplinary Digital Publishing Institute, v. 11, n. 1212, Dec. 2021. ISSN 2073-4395. Disponível em: <https://doi.org/10.3390/agronomy11122578>.

BLASCH, J. et al. Farmer preferences for adopting precision farming technologies: a case study from Italy. **European Review of Agricultural Economics**, 2020. ISSN 0165-1587. Disponível em: <https://doi.org/10.1093/erae/jbaa031>.

BOLFE, E. L. et al. Precision and Digital Agriculture: Adoption of Technologies and Perception of Brazilian Farmers. **Agriculture**, Multidisciplinary Digital Publishing Institute, v. 10, n. 1212, 2020. Disponível em: <https://doi.org/10.3390/agriculture10120653>.

CANDIAGO, S. et al. Evaluating Multispectral Images and Vegetation Indices for Precision Farming Applications from UAV Images. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 7, n. 44, p. 4026–4047, 2015. Disponível em: <https://doi.org/10.3390/rs70404026>.

CHAURASIA, A.; CULURCIELLO, E. LinkNet: Exploiting encoder representations for efficient semantic segmentation. In: **2017 IEEE Visual Communications and Image Processing (VCIP)**. IEEE, 2017. Disponível em: <https://doi.org/10.1109/VCIP.2017.8305148>.

DELAVARPOUR, N. et al. A Technical Study on UAV Characteristics for Precision Agriculture Applications and Associated Practical Challenges. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 13, n. 66, 2021. Disponível em: <https://doi.org/10.3390/rs13061204>.

DOHA, R. et al. Deep Learning based Crop Row Detection with Online Domain Adaptation. In: **Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining**. Association for Computing Machinery, 2021. (KDD '21), p. 2773–2781. ISBN 978-1-4503-8332-5. Disponível em: <https://doi.org/10.1145/3447548.3467155>.

EBRAHIMI, M. S.; ABADI, H. K. Study of Residual Networks for Image Recognition. In: ARAI, K. (Ed.). **Intelligent Computing**. Springer International Publishing, 2021. (Lecture Notes in Networks and Systems), p. 754–763. ISBN 978-3-030-80126-7. Disponível em: <https://doi.org/10.1007/978-3-030-80126-7_53>.

FANG, X. Understanding deep learning via backtracking and deconvolution. **Journal of Big Data**, v. 4, n. 1, 2017. ISSN 2196-1115. Disponível em: <https://doi.org/10.1186/s40537-017-0101-8>.

FAOSTAT. **Food and Agriculture Organization, Corporate Statistical Database (FAOSTAT) - Crops and livestock products**. 2022. Disponível em: <http://www.fao.org/faostat/en/#data/QCL>. Acesso em: 27 de março de 2023.

FAWAKHERJI, M. et al. Crop and Weeds Classification for Precision Agriculture Using Context-Independent Pixel-Wise Segmentation. In: **2019 Third IEEE International Conference on Robotic Computing (IRC)**. IEEE, 2019. p. 146–152. Disponível em: <https://doi.org/10.1109/IRC.2019.00029>.

GARCÍA-SANTILLÁN, I. D. et al. Automatic detection of curved and straight crop rows from images in maize fields. **Biosystems Engineering**, v. 156, p. 61–79, 2017. ISSN 1537-5110. Disponível em: <https://doi.org/10.1016/j.biosystemseng.2017.01.013>.

GONZALEZ, R. C.; WOODS, R. E. **Processamento Digital de Imagens**. 3. ed. São Paulo: Pearson Universidades, 2010. ISBN 978-85-7605-401-6.

HAO, S.; ZHOU, Y.; GUO, Y. A Brief Survey on Semantic Segmentation with Deep Learning. **Neurocomputing**, v. 406, p. 302–321, 2020. ISSN 0925-2312. Disponível em: <https://doi.org/10.1016/j.neucom.2019.11.118>.

HASAN, A. S. M. M. et al. A survey of deep learning techniques for weed detection from images. **Computers and Electronics in Agriculture**, v. 184, 2021. ISSN 0168-1699. Disponível em: <https://doi.org/10.1016/j.compag.2021.106067>.

HASSANEIN, M.; KHEDR, M.; EL-SHEIMY, N. Crop row detection procedure using low-cost UAV imagery system. In: **The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences**. Copernicus GmbH, 2019. XLII-2-W13, p. 349–356. ISSN 1682-1750. Disponível em: <https://doi.org/10.5194/isprs-archives-XLII-2-W13-349-2019>.

IAKUBOVSKII, P. **Segmentation Models**. GitHub, 2019. Disponível em: <https://github.com/qubvel/segmentation_models>. Acesso em: 1 de abril de 2023.

INFOROW. **Inforow - Análise de forma inteligente**. 2021. Disponível em: <https://inforow.com.br/>. Acesso em: 21 de março de 2023.

JIMENEZ, S.; GONZALEZ, F. A.; GELBUKH, A. Mathematical properties of soft cardinality: Enhancing Jaccard, Dice and cosine similarity measures with element-wise distance. **Information Sciences**, v. 367–368, p. 373–389, 2016. ISSN 0020-0255. Disponível em: <https://doi.org/10.1016/j.ins.2016.06.012>.

KAUR, S.; SAHAMBI, J. Cell detection in very low contrast images using Discrete Curvelet Transform and radon transform with morphological operations. In: **2015 2nd International Conference on Recent Advances in Engineering Computational Sciences (RAECS)**. IEEE, 2015. Disponível em: <https://doi.org/10.1109/RAECS.2015.7453419>.

KURUVILLA, J. et al. A review on image processing and image segmentation. In: **2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)**. IEEE, 2016. p. 198–203. Disponível em: <https://doi.org/10.1109/SAPIENCE.2016.7684170>.

LATEEF, F.; RUICHEK, Y. Survey on semantic segmentation using deep learning techniques. **Neurocomputing**, v. 338, p. 321–348, 2019. ISSN 0925-2312. Disponível em: <https://doi.org/10.1016/j.neucom.2019.02.003>.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. IEEE, 2015. p. 3431–3440. ISSN 1063-6919. Disponível em: <https://doi.org/10.1109/CVPR.2015.7298965>.

LU, N. et al. Improved estimation of aboveground biomass in wheat from RGB imagery and point cloud data acquired with a low-cost unmanned aerial vehicle system. **Plant Methods**, BioMed Central, v. 15, n. 11, 2019. ISSN 1746-4811. Disponível em: <https://doi.org/10.1186/s13007-019-0402-3>.

MARQUES FILHO, H.; VIEIRA NETO, O. **Processamento Digital De Imagens**. 1. ed. Rio de Janeiro: Brasport, 1999. ISBN 978-85-7452-009-4.

MAY, A.; RAMOS, N. P. Uso de gemas individualizadas de cana-de-açúcar para a produção de mudas. **Circular técnica: Embrapa**, Jaguariúna, SP, 2019. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/197206/1/Andre-May-CT-29.pdf>. Acesso em: 7 de março de 2023.

MESSINA, G. et al. A Comparison of UAV and Satellites Multispectral Imagery in Monitoring Onion Crop. An Application in the 'Cipolla Rossa di Tropea' (Italy).

Remote Sensing, Multidisciplinary Digital Publishing Institute, v. 12, n. 2020, 2020. Disponível em: <https://doi.org/10.3390/rs12203424>.

MINAEE, S. et al. Image Segmentation Using Deep Learning: A Survey. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, 2021. ISSN 1939-3539. Disponível em: <https://doi.org/10.1109/TPAMI.2021.3059968>.

MOLIJN, R. A. et al. Sugarcane Productivity Mapping through C-Band and L-Band SAR and Optical Satellite Imagery. **Remote Sensing**, Multidisciplinary Digital Publishing Institute, v. 11, n. 99, 2019. Disponível em: <https://doi.org/10.3390/rs11091109>.

MONTEIRO, A. A. O.; WANGENHEIM, A. von. **Orthomosaic Dataset of RGB aerial Images for Weed Mapping**. Brasil: LAPIX/UFSC, 2019. Disponível em: <https://lapix.ufsc.br/weed-mapping-sugar-cane>. Acesso em: 5 de março de 2023.

OGUNSOLA, A.; ZANCANER, A.; SEED, B. **The Sugar Series : The Top 10 Sugar Producing Countries in the World**. 2021. Disponível em: <https://www.czarnikow.com/blog/the-sugar-series-the-top-10-sugar-producing-countries-in-the-world>. Acesso em: 29 de março de 2023.

OLIVEIRA, L. A. d.; MIRANDA, J. H. d.; COOKE, R. A. Water management for sugarcane and corn under future climate scenarios in Brazil. **Agricultural Water Management**, v. 201, p. 199–206, 2018. ISSN 0378-3774. Disponível em: <https://doi.org/10.1016/j.agwat.2018.01.019>.

OLIVEIRA, M. P. d. et al. Mapping Gaps in Sugarcane Fields Using UAV-RTK Platform. **Agriculture**, Multidisciplinary Digital Publishing Institute, v. 13, n. 66, 2023. ISSN 2077-0472. Disponível em: <https://doi.org/10.3390/agriculture13061241>.

OLIVEIRA, P. V. d. **Detecção de Linhas e Falhas de Plantio por meio da Associação de um Algoritmo Genético para Multilimiarização à Transformada Discreta de Wavelet e Transformada de Hough Probabilística e como Mobile Cloud Computing pode Auxiliar na Melhoria de Desempenho**. Dissertação (Mestrado) — Universidade Federal de Uberlândia (UFU), 2020. Disponível em: <https://doi.org/10.14393/ufu.di.2020.685>.

OTSU, N. A Threshold Selection Method from Gray-Level Histograms. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 9, n. 1, p. 62–66, 1979. ISSN 2168-2909. Disponível em: <https://doi.org/10.1109/TSMC.1979.4310076>.

PEREIRA JÚNIOR, P. C. et al. Comparison of Supervised Classifiers and Image Features for Crop Rows Segmentation on Aerial Images. **Applied Artificial Intelligence**, Taylor & Francis, v. 34, n. 4, p. 271–291, Feb. 2020. ISSN 0883-9514. Disponível em: <https://doi.org/10.1080/08839514.2020.1720131>.

_____. Comparison of Classical Computer Vision vs. Convolutional Neural Networks for Weed Mapping in Aerial Images. **Revista de Informática Teórica e Aplicada**, v. 27, n. 44, p. 20–33, Dec. 2020. ISSN 2175-2745. Disponível em: <https://doi.org/10.22456/2175-2745.97835>.

PEREIRA JÚNIOR, P. C.; WANGENHEIM, A. von. **Orthomosaic Dataset of RGB aerial Images for Crop Rows Detection**. Brasil: LAPIX/UFSC, 2019. Disponível em: <https://lapix.ufsc.br/crop-rows-sugar-cane/>. Acesso em: 5 de março de 2023.

PONTI, M. A. et al. Everything You Wanted to Know about Deep Learning for Computer Vision but Were Afraid to Ask. In: **2017 30th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)**. IEEE, 2017. p. 17–41. ISBN 978-1-5386-0619-3. Disponível em: <https://doi.org/10.1109/SIBGRAPI-T.2017.12>.

RABAB, S. et al. A template-free machine vision-based crop row detection algorithm. **Precision Agriculture**, v. 22, n. 1, p. 124–153, 2021. ISSN 1573-1618. Disponível em: <https://doi.org/10.1007/s11119-020-09732-4>.

RAWAT, W.; WANG, Z. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. **Neural Computation**, v. 29, n. 9, p. 2352–2449, 2017. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/neco_a_00990>.

RIBEIRO, C. **Drones na mensuração de falhas de plantio em cana-de-açúcar**. 2020. Disponível em: <https://blog.sensix.ag/drones-na-mensuracao-de-falhas-de-plantio-em-cana-de-acucar/>. Acesso em: 17 de março de 2023.

ROCHA, B. M. et al. Automatic detection and evaluation of sugarcane planting rows in aerial images. **Information Processing in Agriculture**, Apr. 2022. ISSN 2214-3173. Disponível em: <https://doi.org/10.1016/j.inpa.2022.04.003>.

_____. Detection of Curved Rows and Gaps in Aerial Images of Sugarcane Field Using Image Processing Techniques. **IEEE Canadian Journal of Electrical and Computer Engineering**, v. 45, n. 3, p. 303–310, Sept. 2022. Disponível em: <https://doi.org/10.1109/ICJECE.2022.3178749>.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In: NAVAB, N. et al. (Ed.). **Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015**. Springer International Publishing, 2015. (Lecture Notes in Computer Science), p. 234–241. ISBN 978-3-319-24574-4. Disponível em: <https://doi.org/10.1007/978-3-319-24574-4_28>.

ROZO, F. A. H. **An automatic georeferenced crop rows generator using aerial high-resolution images for precision agriculture in sugarcane crops**. Dissertação (Mestrado) — Universidad del Valle, 2019. Disponível em: <https://doi.org/10893/15451>.

RUDORFF, B. F. T. et al. Studies on the Rapid Expansion of Sugarcane for Ethanol Production in São Paulo State (Brazil) Using Landsat Data. **Remote Sensing**, Molecular Diversity Preservation International, v. 2, n. 44, p. 1057–1076, 2010. Disponível em: <https://doi.org/10.3390/rs2041057>.

RUSSAKOVSKY, O. et al. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015. Disponível em: <https://doi.org/10.1007/s11263-015-0816-y>.

SEGATO, S. V. et al. **Atualização em produção de cana-de-açúcar**. 1. ed. Piracicaba: Livroceres, 2006.

SILVA, H. G. P. d. F. et al. Detecção de Falhas em Linhas de Plantio em Imagens Obtidas por VANT Utilizando CNN e Operadores Morfológicos. **Colloquium Exactarum**, v. 14, n. 11, p. 22–35, 2022. ISSN 2178-8332. Disponível em: <https://doi.org/10.5747/ce.2022.v14.n1.e382>.

SILVA, R. R. d. **Detection of Sugarcane Crop Rows From UAV Images Using Semantic Segmentation and Radon Transform**. Dissertação (Mestrado) — Universidade Federal de Uberlândia (UFU, 2020. Disponível em: <https://dx.doi.org/10.14393/ufu.di.2020.736>.

SILVA, R. R. d.; ESCARPINATI, M. C.; BACKES, A. R. Sugarcane crop line detection from UAV images using genetic algorithm and Radon transform. **Signal, Image and Video Processing**, 2021. ISSN 1863-1703, 1863-1711. Disponível em: <https://doi.org/10.1007/s11760-021-01908-3>.

SIMONYAN, K.; ZISSERMAN, A. **Very Deep Convolutional Networks for Large-Scale Image Recognition**. arXiv:1409.1556v6, 2015. Disponível em: <https://doi.org/10.48550/arXiv.1409.1556>.

SOARES, G. A.; ABDALA, D.; ESCARPINATI, M. Plantation Rows Identification by Means of Image Tiling and Hough Transform. In: **Proceedings of the 13th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications**. SCITEPRESS - Science and Technology Publications, 2018. p. 453–459. ISBN 978-989-758-290-5. Disponível em: <https://doi.org/10.5220/0006657704530459>.

SOUZA, C. H. W. d. et al. Mapping skips in sugarcane fields using object-based analysis of unmanned aerial vehicle (UAV) images. **Computers and Electronics in Agriculture**, v. 143, p. 49–56, 2017. ISSN 0168-1699. Disponível em: <https://doi.org/10.1016/j.compag.2017.10.006>.

TAHA, A. A.; HANBURY, A. Metrics for evaluating 3D medical image segmentation: analysis, selection, and tool. **BMC Medical Imaging**, v. 15, n. 1, 2015. ISSN 1471-2342. Disponível em: <https://doi.org/10.1186/s12880-015-0068-x>.

WALSH, B. **Are 8 billion people too many - or too few?** 2023. Disponível em: <https://www.vox.com/the-highlight/23436211/overpopulation-population-8-billion-people>. Acesso em: 15 de maio de 2023.

WALTON, J.; MANSA, J.; SCHMITT, K. R. **The 5 Countries That Produce the Most Sugar**. 2022. Disponível em: <https://www.investopedia.com/articles/investing/101615/5-countries-produce-most-sugar.asp>. Acesso em: 25 de março de 2023.

WANG, Z.; JI, S. Smoothed dilated convolutions for improved dense prediction. **Data Mining and Knowledge Discovery**, v. 35, n. 4, p. 1470–1496, 2021. ISSN 1573-756X. Disponível em: <https://doi.org/10.1007/s10618-021-00765-5>.

ZHANG, A. et al. **Dive into Deep Learning**. [s.n.], 2021. Disponível em: <https://d2l.ai/>. Acesso em: 17 de março de 2023.

ZHAO, H. et al. Pyramid Scene Parsing Network. In: **2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. IEEE, 2017. p. 6230–6239. ISSN 1063-6919. Disponível em: <https://doi.org/10.1109/CVPR.2017.660>.

ZHENG, Y.-Y. et al. CropDeep: The Crop Vision Dataset for Deep-Learning-Based Classification and Detection in Precision Agriculture. **Sensors**, v. 19, n. 5, 2019. ISSN 1424-8220. Disponível em: <https://doi.org/10.3390/s19051058>.

Apêndices

APÊNDICE **A**

Tabelas de resultados

Tabela 30 – Resultados do CD médio e desvio padrão obtidos para cada rede durante o treinamento e teste nos *datasets*.

Rede	**Treinamento**	Equação			
		A	B	C	D
U-Net	A	**0,9080** ± 0,0045	0,8851 ± 0,0078	0,8494 ± 0,0150	0,8150 ± 0,0096
	B	0,8181 ± 0,0242	**0,9233** ± 0,0031	0,8922 ± 0,0033	0,7440 ± 0,0247
	C	0,6322 ± 0,0747	0,8847 ± 0,0151	**0,9044** ± 0,0028	0,5288 ± 0,0734
	D	0,8914 ± 0,0063	0,8938 ± 0,0117	0,8634 ± 0,0082	**0,8672** ± 0,0056
LinkNet	A	**0,8907** ± 0,0029	0,8585 ± 0,0116	0,8311 ± 0,0152	0,7905 ± 0,0111
	B	0,7969 ± 0,0249	**0,9030** ± 0,0024	0,8688 ± 0,0034	0,7284 ± 0,0251
	C	0,7133 ± 0,0331	0,8665 ± 0,0131	**0,8835** ± 0,0024	0,6195 ± 0,0405
	D	0,8743 ± 0,0076	0,8653 ± 0,0090	0,8313 ± 0,0088	**0,8358** ± 0,0056
PSPNet	A	**0,8707** ± 0,0075	0,7982 ± 0,0240	0,7393 ± 0,0366	0,7596 ± 0,0083
	B	0,7622 ± 0,0241	**0,8861** ± 0,0081	0,7737 ± 0,0329	0,7599 ± 0,0074
	C	0,5999 ± 0,0626	0,8420 ± 0,0183	**0,8661** ± 0,0065	0,6046 ± 0,0373
	D	0,8587 ± 0,0120	0,6646 ± 0,0764	0,5487 ± 0,0521	**0,8410** ± 0,0079

Tabela 31 – Resultados do CD médio e desvio padrão obtidos ao treinar cada rede nos *datasets* E(N) e testar nos outros *datasets*.

Rede	**Treinamento**	**Teste**			
		A	B	C	D
U-Net	E(200)	0,9049 ± 0,0058	0,9099 ± 0,0034	0,8916 ± 0,0049	0,8422 ± 0,0049
	E(300)	0,9043 ± 0,0039	0,9104 ± 0,0047	0,8955 ± 0,0046	0,8482 ± 0,0141
	E(400)	0,9069 ± 0,0054	0,9137 ± 0,0042	0,8968 ± 0,0051	0,8495 ± 0,0107
	E(500)	0,9096 ± 0,0045	0,9155 ± 0,0019	0,8993 ± 0,0032	0,8547 ± 0,0037
LinkNet	E(200)	0,8821 ± 0,0051	0,8905 ± 0,0042	0,8697 ± 0,0046	0,8090 ± 0,0105
	E(300)	0,8835 ± 0,0034	0,8920 ± 0,0033	0,8740 ± 0,0025	0,8206 ± 0,0091
	E(400)	0,8823 ± 0,0051	0,8936 ± 0,0047	0,8742 ± 0,0026	0,8196 ± 0,0093
	E(500)	0,8849 ± 0,0037	0,8952 ± 0,0030	0,8762 ± 0,0045	0,8238 ± 0,0069
PSPNet	E(200)	0,8653 ± 0,0085	0,8679 ± 0,0124	0,8560 ± 0,0086	0,8158 ± 0,0062
	E(300)	0,8648 ± 0,0061	0,8713 ± 0,0055	0,8523 ± 0,0080	0,8185 ± 0,0044
	E(400)	0,8751 ± 0,0063	0,8779 ± 0,0080	0,8605 ± 0,0083	0,8263 ± 0,0050
	E(500)	0,8748 ± 0,0041	0,8795 ± 0,0064	0,8609 ± 0,0071	0,8304 ± 0,0028