http://journal.unoeste.br/index.php/ce/index
DOI: 10.5747/ce.2022.v14.n1.e382
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 15/12/2021 Revisado: 17/03/2022 Aceito: 22/03/2022

DETECÇÃO DE FALHAS EM LINHAS DE PLANTIO EM IMAGENS OBTIDAS POR VANT UTILIZANDO CNN E OPERADORES MORFOLÓGICOS

FAILURE DETECTION IN ROW CROPS FROM UAV IMAGES USING CNN AND MORPHOLOGICAL OPERATORS

Hélio Gomes Pereira da Silva Filho[1], Francisco Assis da Silva[1], Leandro Luiz de Almeida[1], Danillo Roberto Pereira[1], Mário Augusto Pazoti[1], Almir Olivette Artero[2], Marco Antônio Piteri[2]

[1]Universidade do Oeste Paulista – UNOESTE, Faculdade de Informática de Presidente Prudente – FIPP, Presidente Prudente, SP.
[2]Faculdade de Ciências e Tecnologia, UNESP - Universidade Estadual Paulista Departamento de Matemática e Computação, Presidente Prudente.
E-mail: helio@unoeste.edu.br; chico@unoeste.br; llalmeida@unoeste.br; mario@unoeste.br; danilopereira@unoeste.br; almir.artero@unesp.br; marco.piteri@unesp.br

RESUMO – A população mundial cresce a cada ano, porém, as terras cultiváveis do planeta já estão praticamente todas em uso ou protegidas por leis ambientais. A humanidade precisa encontrar meios de aumentar a produtividade no campo, e uma das formas é fazendo o uso da tecnologia. Este trabalho utiliza recursos computacionais para detectar falhas em linhas de plantio, por meio da análise de imagens de plantações obtidas por VANTs. Na metodologia desenvolvida foram utilizados CNN, operadores morfológicos e um algoritmo para desenhar as linhas de plantio. Com as falhas detectadas, busca-se auxiliar o produtor rural a tomar melhores decisões, aumentar sua produção e diminuir os prejuízos. Os resultados obtidos foram satisfatórios, mas estão intimamente ligados a qualidade da classificação da imagem pela CNN, que apresentou F1 Score em torno de 92%.
Palavras-chave: Visão Computacional, CNN, linhas de plantio, VANT, operador morfológico.

ABSTRACT – The world population grows every year, however, the arable lands of the planet are practically all in use or protected by environmental laws. Humanity needs to find ways to increase productivity in the countryside, and one of the ways is by making use of technology. This paper uses computational resources to detect failures in planting lines, through the analysis of plantation images obtained by UAVs. In the developed methodology, CNN, morphological operators and an algorithm were used to draw the planting lines. With the detected failures, the aim is to help rural producers to make better decisions, increase their production and reduce losses. The results obtained were satisfactory, but are closely linked to the quality of the image classification by CNN, which presented an F1 Score around 92%.
Keywords: Computer Vision, CNN, row crop, UAV, morphological operator.

1. INTRODUÇÃO

Com o constante aumento da demanda mundial por alimentos e matéria-prima proveniente de plantações, produtores rurais buscam aumentar a produtividade por hectare de suas propriedades ao máximo. De acordo com dados da ONU, numa estimativa feita em 2012 (FAO, 2017), a população mundial em 2024 será superior a 8 bilhões de pessoas, e até 2050 será superior a 9,5 bilhões, um crescimento de 13,16% e 34,90% respectivamente. Porém, ao mesmo tempo, praticamente não existem mais terras disponíveis para expansão agrícola no mundo, ou porque simplesmente acabaram, como no caso da China e dos EUA, ou devido a leis de proteção ao meio ambiente (SAATH; FACHINELLO, 2018).

Para atender a demanda, a agricultura mundial em 2050, terá que produzir quase 50% mais comida, ração e biocombustível do que produzia em 2012, sendo que a África subsaariana e o Sul da Ásia terão que produzir mais do que o dobro, e o restante do mundo na média um terço a mais (FAO, 2017). Com um aumento tão significativo na demanda, sem aumentar a disponibilidade de terras produtivas, se torna indispensável o uso de técnicas da Agricultura de Precisão já presentes, como por exemplo, a utilização de implementos agrícolas cada vez mais modernos, o uso de insumos mais eficientes e também a utilização de dados, obtidos por Satélites, VANT (Veículo Aéreo Não Tripulado) e outros equipamentos, que servirão de entradas para sistemas que os analisam e ajudam ao produtor a tomar melhores decisões.

Com a Agricultura de Precisão, o que antes era feito de forma padronizada e homogênea em toda a lavoura, agora pode ser feita de forma localizada, de acordo com a demanda e as necessidades específicas de cada região da lavoura, dando muito mais controle ao produtor. A agricultura de precisão teria pouco significado, se não fosse a veracidade inquestionável de que há uma variabilidade espaço-temporal sem fim no meio ambiente (PIERCE; NOWAK, 1999).

Com a posse de uma variedade grande de informações sobre a lavoura é possível aplicar soluções na quantidade certa, na hora certa, no local certo, com muito mais precisão e eficácia, o que se traduz em redução nos custos com insumos e recursos humanos. Assim, consequentemente, essas informações contribuem para a preservação do meio ambiente ao reduzir o uso de produtos químicos potencialmente nocivos, havendo uma forte relação entre Agricultura de Precisão e o manejo sustentável dos solos. Sabe-se que os recursos naturais são finitos, assim como os insumos agrícolas, e que não se pode aplica-los de maneira uniforme em um meio altamente variável, sem que os excessos sejam carregados aos rios e fontes de água (WERNER, 2007).

Os VANTs são um grande aliado na busca por esses objetivos, ao propiciar, por exemplo, imagens de alta resolução, que são muito superiores, mais facilmente obtidas e mais baratas do que solicitar imagens orbitais. Um dos problemas que tais imagens ajudam a analisar, é a presença de falhas em linhas de plantio. Uma falha em linhas de cana-de-açúcar, por exemplo, é definida como a projeção da distância entre duas canas consecutivas ao longo da linha. Essa distância é medida de centro a centro dos colmos ao nível do solo, de modo que a distância seja maior do que 50 cm (STOLF, 1986).

A maior parte dos problemas ligados a falhas em linhas de plantio, foram introduzidos a partir de métodos ainda pouco aperfeiçoados de mecanização e automação e também erros operacionais, principalmente nos estágios anteriores à semeadura, assim como na semeadura em si (CANAONLINE, 2016).

As falhas são contabilizadas e classificadas individualmente com base em um tamanho mínimo e em termos de porcentagem com relação a área plantada. Com a posse dessas informações, o responsável técnico ou o produtor rural, pode decidir com mais assertividade se realizará alguma ação, como por exemplo, replantio de falhas maiores que um certo tamanho, ou se as informações serão utilizadas para economia de insumos nas linhas de falhas, visto que o replantio nem sempre é uma opção ideal, já que a competição por nutrientes e luz com plantas mais velhas, pode prejudicar o desenvolvimento de plantas mais novas. Estipula-se, por exemplo, que em uma cultura de milho para cada 10% de aumento do Coeficiente de Variação do espaçamento entre sementes, perde-se, 1,5 saco de grãos produzidos por hectare de lavoura plantada (SILVA, 2015).

O processo, de coleta de dados sobre falhas em linhas de plantio, quando é feito de forma manual, tradicionalmente se enviam equipes para o campo, que colhem essas informações por amostragem, o que é um processo com baixa precisão e custoso para o produtor. No trabalho realizado por Alves *et al*.

(2015), foi observado que há uma discrepância muito grande dos resultados mais precisos que são conseguidos através das imagens aéreas e o posterior processamento de imagens com relação ao método manual. Na plantação analisada, a porcentagem de falhas encontradas manualmente foi de 28,03%, e por processamento de imagens chegou-se a 14,76%, concluindo que os métodos manuais são muito genéricos e se utilizam de amostragens aleatórias para depois extrapolar os resultados para todo o hectare, por ser impraticável analisar manualmente toda a extensão da área plantada.

Em Kamilaris e Prenafeta-Boldú (2018) foi verificado que técnicas mais recentes de *Deep learning*, incluindo CNN (*Convolutional Neural Networks*), estão sendo cada vez mais utilizadas em agricultura de precisão, obtendo resultados melhores do que métodos convencionais de processamento de imagens. Essas técnicas estão atingindo maior precisão nos resultados, porém dependem de *datasets* suficientemente grandes (algumas centenas ou milhares de imagens) e de alta qualidade (baixo ruído, alta resolução, alta variação, etc).

Bah, Rafiane e Canals (2019), propuseram um método para identificar linhas de plantio utilizando CNN, que conseguiu atingir uma precisão de 93,58%. Segundo os autores, isso possibilita, por exemplo, que máquinas guiadas por GPS percorram a plantação com o mínimo de danos possível.

Apesar de já existirem algumas soluções possíveis para vários problemas enfrentados em Agricultura de Precisão, ainda existe espaço para melhorias em várias situações. O aumento da precisão com uso de tecnologias de aprendizado de máquina, identificado em outros trabalhos, como em Kamilaris e Prenafeta-Boldú (2018), verifica-se que há um enorme potencial ainda a ser explorado. Diante disso o produtor fica mais suscetível a perdas, por não ter acesso a métodos mais precisos que o auxiliem nas decisões, levando a uma menor produtividade, gastos excessivos em insumos agrícolas e a um baixo aproveitamento da terra em geral.

Este trabalho busca contribuir propondo um método de identificação de falhas em linhas de plantio utilizando imagens aéreas RGB de VANT de baixo custo, utilizando a precisão da classificação de uma CNN, juntamente com processamento digital de imagens e operadores morfológicos. A CNN é utilizada para classificar os pixels da imagem obtida por VANT da plantação, separando em verde a cultura de interesse, em amarelo as ervas daninhas e em vermelho o solo. Com a imagem classificada, foram utilizadas técnicas de processamento de imagem para obter dados que facilitarão todo o processo, como por exemplo, o ângulo em que as linhas de plantio estão dispostas na imagem. Os operadores morfológicos são importantes para evidenciar regiões de falhas em linhas de plantio e para eliminação de ruídos.

Após esta seção introdutória, o trabalho está organizado da seguinte forma. Na Seção 2 são apresentados os conceitos que fundamentam a base teórica para este trabalho. Na seção 3 é detalhado todo o processo desenvolvido para a detecção de falhas em linhas de plantio. Na seção 4 são apresentados os resultados e, por fim, na seção 5 encontram-se as conclusões finais.

2. CONCEITOS FUNDAMENTAIS

A seguir serão apresentados resumidamente os conceitos fundamentais para a compreensão do funcionamento da rede neural convolucional utilizada neste trabalho, que é a PSPNet (ZHAO *et al.*, 2017). Para tal, serão apresentados os fundamentos que a diferenciaram para que ela obtenha resultados melhores em comparação com outras redes do tipo semânticas baseadas em FCN (*Fully Convolutional Network*). A PSPNet é composta por uma CNN do tipo ResNet, com convolução dilatata nas duas camadas mais altas e pelo *Pyramid Pooling Module*.

2.1. ResNet

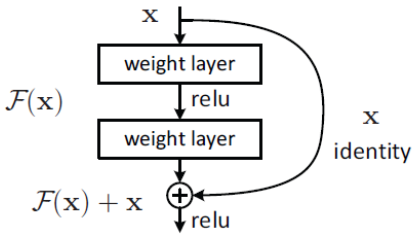
Segundo He *et al.* (2015), os problemas clássicos de desaparecimento e explosão do gradiente em redes neurais profundas já foram praticamente solucionados com a inicialização normalizada e normalização intermediária de camadas. Isso permite a construção de redes neurais com dezenas de camadas que convergem com gradiente descendente estocástico (SGD - *stochastic gradient descent*) e *backpropagation*.

Porém, a partir de um certo ponto, quando a rede já está “suficientemente” profunda, conforme são adicionadas mais camadas à rede, a acurácia é saturada e depois se degrada rapidamente. Esse problema de degradação, segundo He *et al.* (2015), provavelmente não é causado por *overfitting* nem pelo problema de desaparecimento/explosão do gradiente. Os autores apenas constataram a existência de uma

dificuldade da rede em se otimizar conforme se aumenta sua profundidade, porém, a verdadeira causa desse problema ainda deve ser estudada mais a fundo. Resumindo, adicionar mais camadas a uma rede já adequadamente profunda resulta em um aumento no erro de treinamento.

Os autores conseguiram solucionar o problema com a utilização de um bloco de aprendizagem residual, como apresentado na Figura 1. Com esse bloco, fica mais fácil para rede ser otimizada, pois terá uma referência para se ajustar caso a ativação resultante de uma camada seja muito próxima ou igual a zero.

Figura 1. Um bloco de aprendizagem residual.



Fonte: (HE *et al.*, 2015).

Supondo que a entrada x, é a saída da camada anterior ao bloco residual, e l é o nível de profundidade da camada, pode-se expressar isso com a Equação 1.

$$x = a^l \quad (1)$$

Supondo que Z^{l+2} é a função linear onde a matriz W^{l+2} de pesos é aplicada, suprimindo-se o *bias* b^{l+2}para facilitar a representação, e que a função $g(x)$ é a função não-linear ReLU (*Rectified Linear Unit*), que produz resultados iguais ou maiores que zero, pode-se expressar o resultado ao final do bloco residual como na Equação 3.

$$Z^{l+2} = W^{l+2}a^{l+1} \quad (2)$$

$$a^{l+2} = g(W^{l+2}a^{l+1} + a^l) \quad (3)$$

Pode-se observar que se o termo $W^{l+2}a^{l+1}$, na Equação 3, tender a zero, fica muito difícil para a rede encontrar os parâmetros para otimizar a rede, ocorrendo o problema da degradação. Porém, com a adição da referência a^l, se torna muito mais fácil para a rede ser otimizada, num caso extremo em que $W^{l+2}a^{l+1} = 0$, o sinal permanece ainda com o valor de a^l, e, portanto, a rede pelo menos não é degradada e pode até aprender algum parâmetro e aumentar o desempenho da rede.

E por último, no caso em que a^{l+2} tenha dimensões diferentes de a^l, utiliza-se a matriz W_s para ajustar as dimensões de a^l, como na Equação 4. Isso pode ser feito de duas formas, ou se faz um *zero padding* ou a matriz W_s pode conter parâmetros a serem aprendidos pela rede.

$$a^{l+2} = g(W^{l+2}a^{l+1} + W_s a^l) \quad (4)$$

2.2. Convolução Dilatada

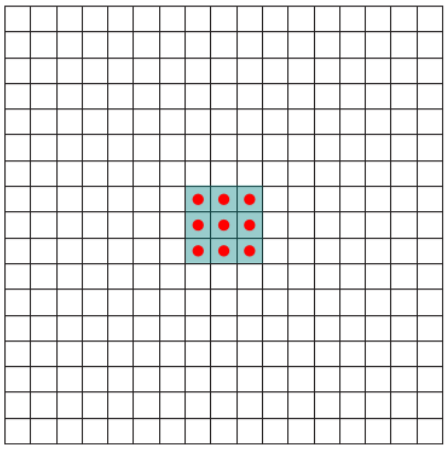
Segundo Yu e Koltun (2016), uma convolução discreta convencional é descrita pela Equação 5, e uma convolução l-dilatada pela Equação 6. Na Figura 2 são descritos exemplos de utilização da convolução dilatada e sua influência no aumento do campo receptivo. A convolução dilatada suporta a expansão exponencial do campo receptivo sem perda de resolução ou cobertura. O número de parâmetros associados a cada camada é idêntico. O campo receptivo cresce exponencialmente, enquanto o número de parâmetros cresce linearmente.

$$(F * k)(p) = \sum_{s+t=p} F(s)k(t) \quad (5)$$

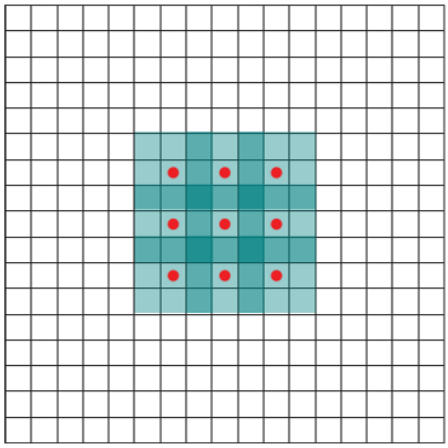
$$(F *_l k)(p) = \sum_{s+lt=p} F(s)k(t) \quad (6)$$

Figura 2. Exemplos de utilização da convolução dilatada para aumentar o campo receptivo. a) F_1 é produzido a partir de F_0 por uma convolução 1-dilatada; cada elemento em F_1 tem um campo receptivo de 3x3. b) F_2 é produzido a partir de F_1 por uma convolução 2-dilatada; cada elemento em F_2 tem um campo receptivo de 7x7. c) F_3 é produzido a partir de F_2 por uma convolução 4-dilatada; cada elemento em F_3 tem um campo receptivo de 15x15.

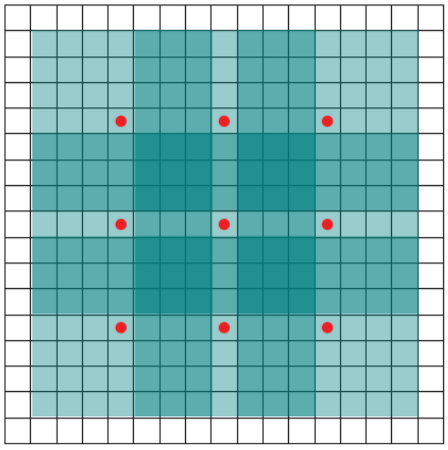
a)



b)



c)



Fonte: (YU; KOLTUN, 2016).

A Convolução dilatada aumenta o campo receptivo de cada valor da matriz resultante sem aumentar a quantidade de parâmetros necessários e, portanto, sem aumentar o custo computacional. Cada valor da matriz resultante será influenciado por um campo maior da matriz/imagem de entrada. Isso na prática significa que haverá maior influência do contexto.

2.3. PSPNet

De acordo com Zhao *et al.* (2017), a motivação para criar uma rede neural do tipo semântica, para análise de cenas aprimorada, foi principalmente os erros de classificação por falta de correlação com o contexto dos pixels classificados. Na sequência segue um resumo dos problemas comuns encontrados em redes FCN - *Fully Convolutional Network* (LONG; SHELHAMER; DARRELL, 2015), que é a rede usada geralmente como base para redes do tipo semânticas.

Relacionamento incompatível - A relação de contexto é universal e importante, especialmente para a compreensão de cenas complexas. Existem padrões visuais co-ocorrentes. Por exemplo, é provável que um avião esteja na pista ou voe no céu, mas não sobre uma estrada. Uma FCN pode classificar um barco como um carro com base na sua aparência. Mas o conhecimento comum é que raramente um carro passa sobre um rio. A falta de capacidade de coletar informações contextuais aumenta a chance de classificação incorreta.

Categorias ambíguas – Segundo Zhou *et al.* (2017) existem muitos pares de rótulos de classe no conjunto de dados ADE20K, que são confusos na classificação. O ADE20K é um *dataset* público, muito usado para fazer *benchmarks* de redes neurais convolucionais semânticas. Os exemplos são campo e terra; montanha e colina; parede e casa; edifício e arranha-céus. Eles possuem aparência semelhante. O anotador especialista que rotulou todo o conjunto de dados ainda comete 17,60% de erro de pixel. A FCN pode por exemplo, prever um objeto como parte arranha-céu e parte edifício. Esses resultados devem ser excluídos para que todo o objeto seja um arranha-céu ou um edifício, mas não ambos. Esse problema pode ser remediado utilizando a relação entre as categorias.

Classes imperceptíveis - A cena contém objetos de tamanho arbitrário. Várias coisas de tamanho pequeno, como postes de luz e letreiros, são difíceis de encontrar, embora possam ser de grande importância. Ao contrário, grandes objetos ou coisas podem exceder o campo receptivo do FCN e, assim, causar predição descontínua.

Para melhorar o desempenho de objetos notavelmente pequenos ou grandes, deve-se prestar muita atenção às diferentes sub-regiões que contêm coisas de categoria imperceptível. Para resumir essas observações, muitos erros estão parcial ou totalmente relacionados ao relacionamento contextual e às informações globais para diferentes campos receptivos. Assim, uma rede profunda com um nível anterior de

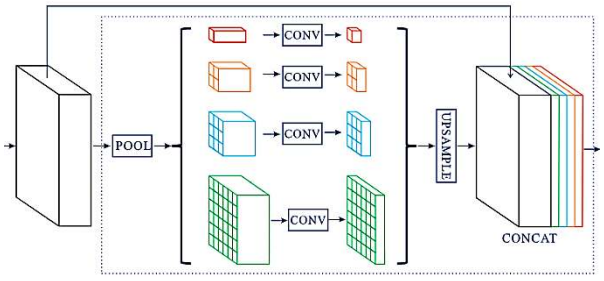
cena global adequado pode melhorar muito o desempenho da análise de cena.

Segundo Zhao *et al.* (2017), esses problemas foram resolvidos ou minimizados, resumidamente da seguinte forma: a imagem de entrada passa por uma CNN do tipo ResNet com convolução dilatada nas últimas duas camadas; os *feature maps* resultantes passam pelo *Pyramid Pooling Module*, e o resultado é concatenado com os *feature maps*; finalmente o resultado da concatenação passa por uma última camada convolucional e é obtida a predição final por pixel. Levando-se em consideração que um *feature map* é o resultado da função de ativação sobre a convolução de um filtro de uma camada da rede neural convolucional com a sua entrada ou a saída da camada anterior, e que em uma camada geralmente existem vários filtros, cada um identificando uma característica diferente que foi aprendida, resultando em vários *feature maps* por camada, que servirão de entrada para a próxima camada.

2.4. Pyramid Pooling Module

Zhao *et al*. (2017) apresentam o módulo como mostrado na Figura 3, que tem por objetivo minimizar os erros já apresentados anteriormente (Seção 2) em redes baseadas em FCN.

Figura 3. Esquema do *Pyramid Pooling Module*, que recebe como entrada os *feature maps* de uma CNN do tipo ResNet com convolução dilatada, obtendo informações importantes de contextos globais e sub-regionais.



Fonte: (ZHAO *et al.*, 2017).

Segundo Zhao *et al.* (2017), uma rede do tipo ResNet, teoricamente, tem um campo receptivo maior do que a imagem de entrada, porém, na prática o campo receptivo é muito menor do que o teórico nas camadas mais altas.

Para reduzir ainda mais a perda de informações de contexto entre diferentes sub-regiões, Zhao *et al.* (2017) propuseram um precedente global hierárquico, contendo informações com diferentes escalas e variando entre as diferentes sub-regiões.

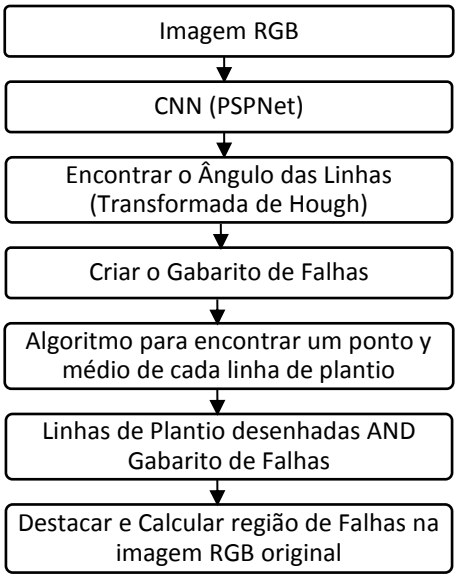
O *Pyramid Pooling Module* funde *features* em quatro escalas diferentes da pirâmide. O nível mais grosseiro destacado em vermelho (Figura 3) é o agrupamento global para gerar uma saída de valor único para cada mapa. Os níveis seguintes da pirâmide separam o *feature map* em diferentes sub-regiões e forma uma representação combinada para diferentes locais. Para manter o peso do contexto global, é usado uma camada de convolução 1x1 após cada nível da pirâmide para reduzir a dimensão da representação do contexto para 1/N do original, com o número de níveis da pirâmide igual a N. Em seguida, é feito o *upsample* diretamente dos *feature maps* de baixa dimensão para obter o mesmo tamanho do *feature map* original por meio de interpolação bilinear. Finalmente, diferentes níveis de mapas são concatenados como o contexto global final. O *Pyramid Pooling Module* usado possui 4 níveis de pooling: 1x1, 2x2, 3x3 e 6x6.

3. MÉTODO PROPOSTO

Para a solução proposta, foi utilizada a linguagem Python com as bibliotecas OpenCV (2021), NumPy (2021) e TensorFlow (2021). De maneira resumida, o fluxo geral de execução começa com o pré-processamento de um segmento de imagem de tamanho 2048x512 pixels dos ortomosaicos RGB, contidos nos *datasets* apresentados nas Figuras 5 e 6, por meio de sua classificação pela CNN PSPNet (ZHAO *et al.*, 2017). Após a classificação o segmento é filtrado para que permaneça apenas a cultura de interesse, no caso a cana-de-açúcar em verde como mostrado na Figura 5, e são eliminadas as ervas daninhas classificadas em amarelo. A imagem filtrada passa então pelo módulo de identificação do ângulo das linhas, que utiliza a Transformada de Hough (1962). Sabendo-se o ângulo, a imagem classificada passa pelo módulo de criação do gabarito de falhas que utiliza Operadores Morfológicos. De posse de todos os dados anteriores, a imagem classificada passa pelo módulo de identificação das linhas de plantio. E finalmente são feitos os cruzamentos dessas informações para obter a imagem final com as falhas identificadas.

O fluxograma da Figura 4 apresenta a sequência das etapas do método proposto.

Figura 4. Fluxograma de processos da metodologia desenvolvida neste trabalho.



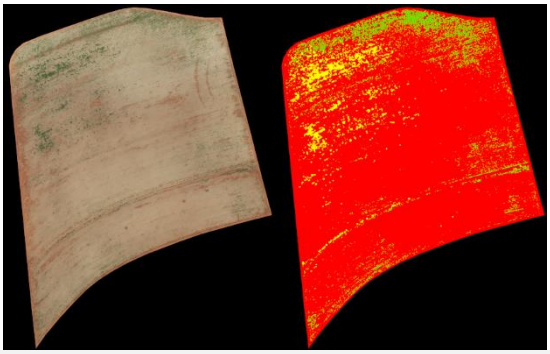
Fonte: Os autores.

3.1. Dataset, Treinamento e Classificação da CNN

Foram considerados dois *datasets*, com ortomosaicos RGB e seus respectivos *Ground Truths* (GT) de duas plantações de cana-de-açúcar distintas do norte do Brasil. O *Ground Truth* em *machine learning,* é a verdadeira natureza do problema, é o alvo que um modelo de aprendizado de máquina deve aprender a reconhecer via aprendizado supervisionado. Em redes semânticas, um *Ground Truth* se refere a uma imagem do *dataset* rotulada, com cada pixel anotado de acordo com a classe que pertence, por exemplo, na Figura 5(b) contém o *Ground Truth* da Figura 5(a), com os rótulos definidos para cada pixel da imagem, em vermelho para o solo, em verde para a cana-de-açúcar e em amarelo para ervas daninhas.

Ambos *datasets* foram construídos com imagens obtidas por um VANT de asas-fixas da Horus Aeronaves, equipado com uma câmera de espectro visível RGB Canon G9X, numa altitude entre 125 a 200 metros, com resolução espacial aproximada de 5cm/pixel. O primeiro *dataset* contêm ervas daninhas, porém, com muito solo exposto. Esse *dataset* foi disponibilizado por Monteiro e Wangenheim (2019) com resolução de 5364x6815 pixels, apresentado na Figura 5. Também foi utilizado um *dataset* com a cultura de interesse bem mais presente, disponibilizado por Pereira Júnior e Wangenheim (2019) com resolução de 9391x6595 pixels, apresentado na Figura 6. Ambos disponíveis para download pelo LAPiX (*Image Processing and Computer Graphics Lab*) da UFSC (Universidade Federal de Santa Catarina).

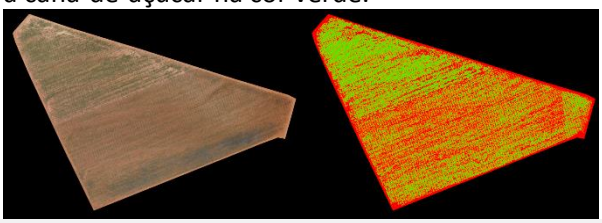
Figura 5. *Dataset* contendo ervas daninhas, porém com muito solo exposto. Em a) o ortomosaico RGB e em b) o seu *Ground Truth*. O solo está na cor vermelha, a cana-de-açúcar na cor verde e as ervas daninhas na cor amarela.



a) b)

Fonte: (MONTEIRO; WANGENHEIM, 2019).

Figura 6. *Dataset* sem ervas daninhas porém balanceado. Em a) o ortomosaico RGB e em b) o seu *Ground Truth*. O solo está na cor vermelha e a cana-de-açúcar na cor verde.



a) b)

Fonte: (PEREIRA JÚNIOR; WANGENHEIM, 2019).

Para o treinamento da PSPNet, os ortomosaicos foram divididos em imagens quadradas de 256x256 pixels. Antes de separar em conjuntos de treinamento, validação e teste, foi feita uma limpeza para evitar que o conjunto fique desbalanceado e ocorra *overfitting*. As imagens com mais de 55% de solo (cor vermelha) ou pixels pretos (cantos e bordas inclinadas do ortomosaico) foram descartadas. Depois de feita a limpeza, foram separadas de forma aleatória 60% das imagens para treinamento, 20% para validação e 20% para teste.

O treinamento foi feito em um PC Desktop com um Intel Core i9 9900k, RTX 2080 Ti 11GB GDDR6 e 64GB de memória RAM DDR4 3600 Mhz. O treinamento levou em torno de quatro horas e 200 épocas para obter um F1 Score de aproximadamente 92%. Na Figura 7 pode-se observar dois exemplos de predições feitas pela CNN.

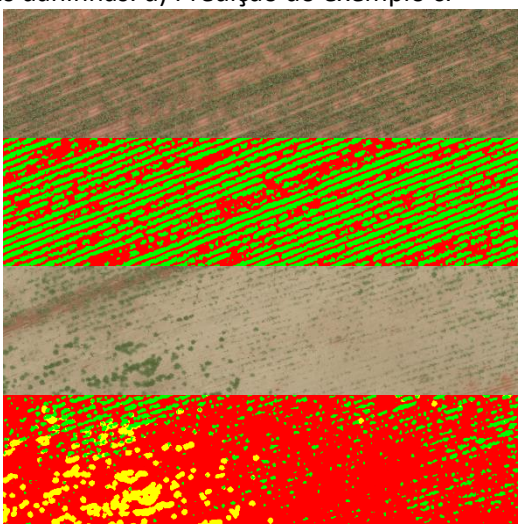
Figura 7. Dois exemplos de predição: a) Sem ervas daninhas. b) Predição do exemplo a. c) Com ervas daninhas. d) Predição do exemplo c.

a)

b)

c)

d)



Fonte: Os autores.

3.2. Encontrar o ângulo das linhas de plantio

No método proposto, foi necessário determinar qual o ângulo em que as linhas de plantio se encontram na imagem para que, se necessário, seja feita a rotação sem perdas da imagem para que as linhas fiquem dispostas horizontalmente para dar continuidade no processo.

Para fazer isso, como apresentado no exemplo da Figura 8, primeiramente foi feita a filtragem da imagem classificada pela CNN, para eliminar os pixels identificados como ervas daninhas, ou seja, que estejam na cor amarela. Após a filtragem, a imagem é convertida do espaço de cor RGB para tons de cinza. Depois, aplica-se o algoritmo de Otsu (1979), que encontra o limiar mais adequando para realizar a binarização. Com a imagem já binarizada, utiliza-se o algoritmo de Zhang e Suen (1984), que faz a esqueletização da imagem.

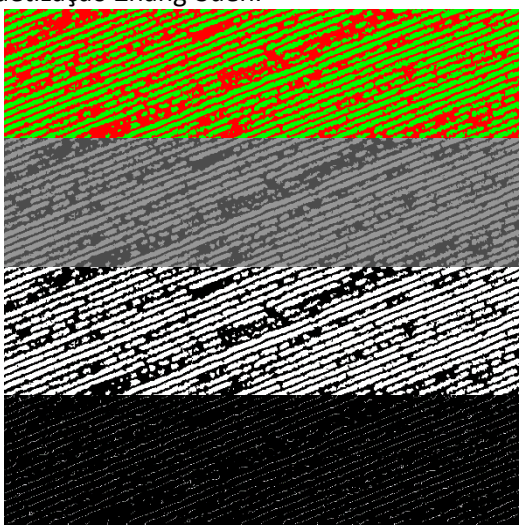
Figura 8. Etapas até a esqueletização. a) Predição filtrada. b) Tons de cinza. c) Binarização Otsu. d) Esquetização Zhang-Suen.

a)

b)

c)

d)



Fonte: Os autores.

Neste ponto já é possível aplicar a Transformada de Hough para Linhas, porém, em alguns casos ainda foi necessário fazer uma limpeza adicional. A limpeza consiste em eliminar ruídos que podem atrapalhar o processo, eliminando distorções indesejadas causadas pela esqueletização, como mostrado no exemplo da Figura 9. As distorções consistem em pequenos fragmentos de linha ao redor da imagem, que lembram uma borda, e pequenas protuberâncias, como se fossem galhos em um tronco de uma árvore, que por estarem em diversos ângulos diferentes, podem causar erros na identificação do ângulo correto.

Figura 9. Exemplo de ruídos que devem ser eliminados.

a)

b)



Fonte: Os autores.

Na Figura 9(a) é possível verificar uma das distorções indesejadas causada pelo algoritmo de

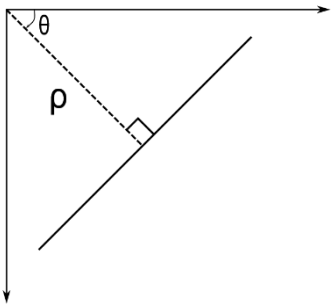
esqueletização de Zhang-Suen, que são segmentos de linhas retas de 1 pixel de espessura, localizados por toda a região da borda da imagem. Essa distorção é corrigida simplesmente desenhando um retângulo com lados de 1 pixel de espessura na cor preta de forma que todos esses segmentos de reta são eliminados da borda da imagem.

Já na Figura 9(b) verifica-se distorções como ramificações ou "galhos" ou "apêndices" que se formam ao longo das linhas de plantio, e que são distorções comuns da esqueletização de Zhang-Suen. Para eliminar essas ramificações de uma forma mais eficiente, é preciso, inicialmente encontrar o ângulo em que as linhas de plantio estão, e, portanto, serão eliminadas posteriormente durante o processo utilizado para desenhar as linhas de plantio, que é quando uma imagem mais semelhante a linhas paralelas, será necessária.

Para encontrar o ângulo desejado, aplica-se a Transformada de Hough para Linhas, que detecta retas em uma imagem na forma da equação paramétrica, apresentada na Equação 7, e visualmente representada na Figura 10.

$$\rho = xcos\theta + ysin\theta \quad (7)$$

Figura 10. Representação visual da equação paramétrica da reta.

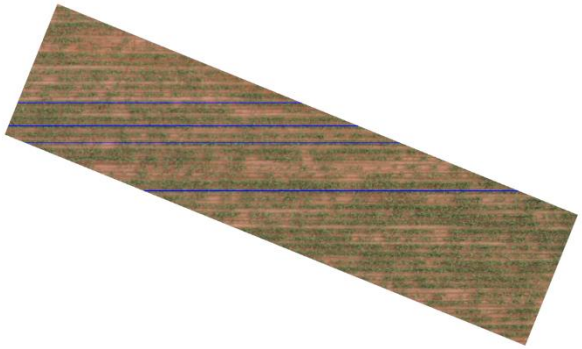


Fonte: (HOUGH, 2016).

Utilizando-se a função HoughLines da biblioteca OpenCV, é possível filtrar as linhas detectadas e selecionar a(s) linha(s) mais bem votada(s), ou seja, a linha mais provável de existir na imagem, de onde é obtido o parâmetro θ em radianos, como demonstrado nas linhas destacadas no exemplo da Figura 11. Para ficar mais claro, é convertido o ângulo de radianos para graus, e então é calculado o ângulo de rotação necessário para que a reta fique disposta horizontalmente, ou seja com o parâmetro θ igual a 90^o, como na Equação 8.

$$\theta_{rotação} = 90^o - \theta \quad (8)$$

Figura 11. Exemplo de uma imagem rotacionada de modo que as linhas de plantio fiquem na posição horizontal, ou seja, paralelas ao eixo *x*, ou eixo das abscissas.



Fonte: Os autores.

3.3. Criar o Gabarito de Falhas

Após o ângulo correto de rotação ter sido detectado, para que as linhas de plantio fiquem paralelas ao eixo das abscissas, o valor deste ângulo θ é guardado em memória RAM, para ser utilizado durante as próximas etapas.

O próximo passo é ler a predição feita pela CNN em tons de cinza e realizar a rotação utilizando o ângulo já calculado anteriormente. O objetivo nesta etapa é evidenciar as regiões de falhas, para servir como fundamento para o restante do processo, e também eliminar ruídos e falhas muito pequenas.

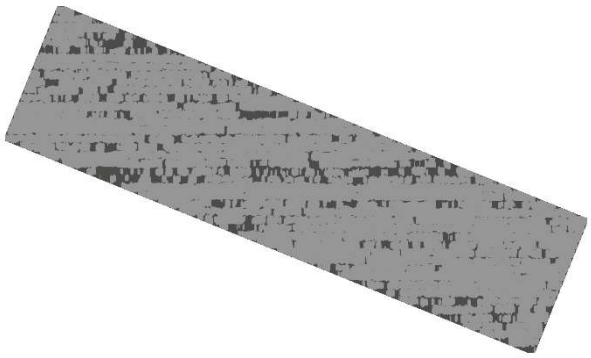
A predição em tons de cinza, como no exemplo da Figura 8(b), é rotacionada utilizando-se o ângulo θ encontrado. Em seguida é realizada uma operação de morfologia matemática de dilatação, utiliza-se um elemento estruturante alongado verticalmente, cujo exemplo de resultado está na Figura 12. O elemento utilizado foi um retângulo 1x15 pixels com a origem no centro.

O valor de 15 pixels foi obtido empiricamente, mas pode ser realizado por um algoritmo, que tem que calcular dois valores, um é a média da distância entre as linhas de plantio e o outro é a média da altura da região verde. A média das distâncias entre as linhas de plantio, pode ser calculada utilizando o vetor de pontos $\overline{y}$, que é citado na Seção 3.4 e demonstrado na Figura 17. A altura média da região verde, pode ser calculada fazendo a varredura vertical ao longo de toda a predição já rotacionada e ao encontrar um pixel verde depois de um pixel vermelho é iniciada a contagem, até encontrar outro pixel vermelho. O valor encontrado é

adicionado a uma variável, e um contador é incrementado, para no final obter-se a média. No exemplo da Figura 12, a subtração destes dois valores resultou em 7 pixels, que em um elemento estruturante retangular com origem no centro representa-se por 1x15 pixels.

Com isso é garantido que os pixels da cultura de interesse não irão se sobrepor, e apenas a região de solo será preenchida, para que as regiões de falhas fiquem em evidência.

Figura 12. Exemplo com predição depois da operação de dilatação.

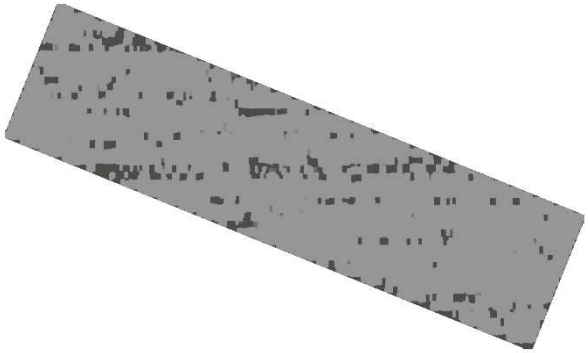


Fonte: Os autores.

Depois da operação de dilatação ter sido aplicada, a imagem resultante é muito ruidosa e com algumas regiões de falha muito pequenas, que podem ser desprezadas. Para melhorar o resultado é aplicada uma operação morfológica de fechamento seguida de uma operação de abertura. Em ambas foi utilizado um elemento estruturante quadrado de 9x9 pixels com origem no centro, com resultado mostrado na Figura 13.

Utilizar uma operação de fechamento, seguida de uma operação de abertura é muito comum para eliminar pequenos "buracos" indesejados na imagem. A operação de fechamento consiste em uma operação de morfologia matemática do tipo dilatação seguida de uma erosão, geralmente utilizando-se o mesmo elemento estruturante para se manter fiel a parte não ruidosa da imagem. Já a operação de abertura consiste em uma operação de morfologia matemática do tipo erosão seguida de uma dilatação.

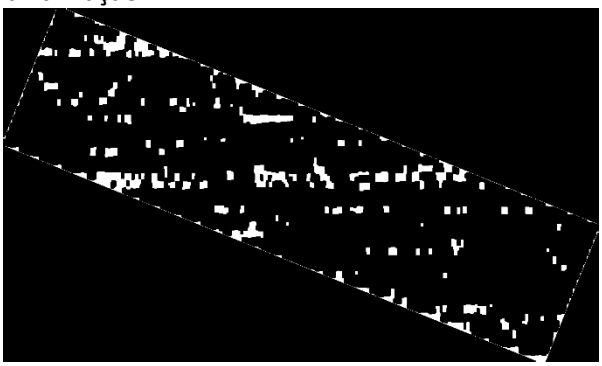
Figura 13. Exemplo depois da operação de fechamento seguida de uma operação de abertura.



Fonte: Os autores.

O resultado dessas operações de morfologia matemática, demonstrado no exemplo da Figura 13, é então binarizada para ser utilizada como uma máscara ou gabarito de falhas, que auxiliará, posteriormente, a demarcação das falhas nas linhas de plantio. Para isso, é aplicado o algoritmo de Otsu (1979), e o resultado pode ser verificado no exemplo da Figura 14.

Figura 14. Exemplo do resultado final, após a binarização.



Fonte: Os autores.

Na Figura 14, é possível verificar que existem ruídos e distorções nas bordas da imagem, porém, são pixels que estão além do tamanho original da imagem. Ao final, a rotação será desfeita, e estes ruídos serão eliminados, pois a imagem será retornada às dimensões originais.

3.4. Desenhar Linhas de Plantio

Após ser obtida a imagem, que foi chamada de gabarito de falhas, é interessante se obter com precisão o ponto *y* central de cada linha de plantio, para que seja possível posteriormente destacar claramente as regiões de falha na imagem original. Para isso, foi um

criado um algoritmo que calcula o ponto *y* médio de cada linha de plantio na imagem rotacionada. Mas antes de executá-lo é preciso realizar uma limpeza adicional na imagem esqueletizada Figura 8(d), para eliminar os ruídos mostrados na Figura 9(b).

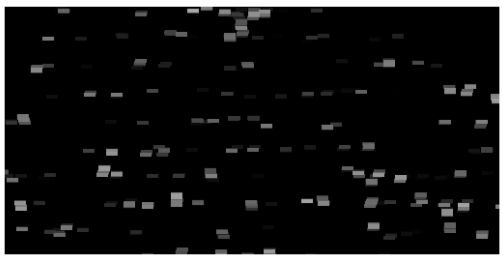
O processo para eliminar esses pequenos "apêndices", mostrados na Figura 9(b), é realizado em três passos. Primeiramente é aplicada uma operação de abertura com um elemento retangular levemente alongado verticalmente, na imagem esqueletizada (Figura 8(d)), de dimensões 1x5 pixels e origem no centro, resultando no exemplo da Figura 15(a). Com a abertura, os apêndices ficaram evidentes, o segundo passo é reforçar sua evidencia, aplicando uma dilatação, com um elemento estruturante retangular alongado horizontalmente de dimensões 11x1 pixels e com origem no centro, resultando no exemplo da Figura 15(b).

Figura 15. Exemplo do processo para evidenciar os apêndices indesejados. a) Região após a abertura. b) Região após a dilatação horizontal.

a)



b)



Fonte: Os autores

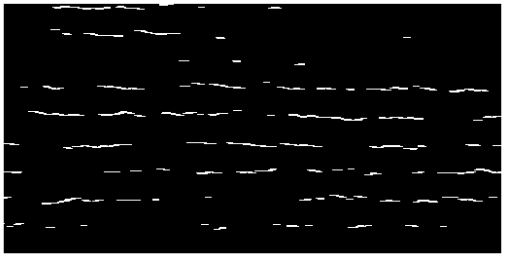
Com os apêndices bem evidentes, o terceiro passo é aplicar uma operação XOR (Exclusive OR), do resultado demonstrado no exemplo da Figura 15(b) com o resultado da esqueletização demonstrado no exemplo da Figura 8(d). O resultado da operação XOR, demonstrado na Figura 16(a), já é mais semelhante com linhas horizontais paralelas, porém, ainda é possível se observar um pouco de ruído remanescente, que são alguns pixels nas regiões entre uma linha e outra. Para deixar a imagem ainda mais limpa, ou seja, com a maior parte dos pixels muito próximos do centro de massa das linhas de plantio, é aplicada mais uma operação de abertura com um elemento estruturante retangular de dimensões 3x1 pixels com origem no centro, resultando no exemplo da Figura 16(b).

Figura 16. Exemplo de imagem esqueletizada após as operações para eliminar excesso de ruídos. a) Após a operação XOR. b) Após a abertura.

a)



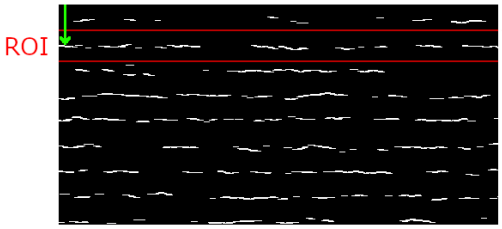
b)



Fonte: Os autores.

O resultado demonstrado no exemplo da Figura 16(b) já está bom o suficiente para dar continuidade ao processo de detecção das linhas de plantio. É feita uma varredura vertical, como exemplificado na Figura 17, ao longo de toda a imagem, de modo que cada pixel diferente de zero encontrado, verifica-se o valor *y* de sua posição. Se o valor de *y* estiver dentro de uma região de interesse (ROI) já contida num vetor de médias, o valor *y* é adicionado na posição encontrada e a média do vetor é recalculada, senão é criada uma nova posição com uma nova região de interesse, indicando que se trata de uma nova linha de plantio.

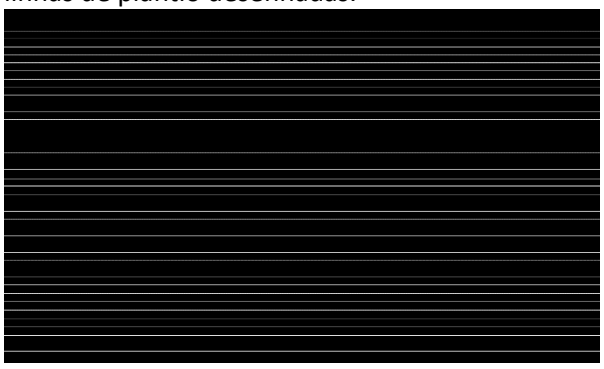
Figura 17. Exemplo de funcionamento do algoritmo que detecta onde desenhar as linhas de plantio. Com a varredura vertical é encontrado um ponto diferente de zero, como é a primeira linha encontrada, é criada uma região de interesse pra esta linha, onde todos as posições *y* dos pixels nessa região serão usadas para calcular a posição *y* média desta linha. E o processo continua para todas as linhas.



Fonte: Os autores.

De posse do vetor com as posições *y* médias de cada linha de plantio é criada uma nova imagem, com as mesmas dimensões da imagem rotacionada, e utilizadas as posições *y* médias encontradas pelo algoritmo, as linhas são desenhadas na nova imagem, resultando no exemplo da Figura 18.

Figura 18. Exemplo de imagem criada com as linhas de plantio desenhadas.



Fonte: Os autores.

3.5. Calcular e Destacar as falhas

De posse do gabarito de falhas, como demonstrado no exemplo da Figura 14, e da imagem com as linhas de plantio desenhadas, como no exemplo da Figura 18, é aplicada uma operação AND entre essas duas imagens, obtendo-se os pixels referentes as falhas detectadas nas linhas de plantio, resultando no exemplo da Figura 19. Esses pixels são contabilizados e a partir da resolução espacial da imagem se deduz o valor em centímetros ou metros.

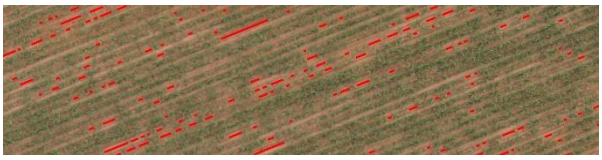
Figura 19. Exemplo de falhas detectadas.



Fonte: Os autores..

Com esse resultado, como demonstrado no exemplo da Figura 19, pode-se desenhar as falhas de forma destacada na imagem original e desfazer a rotação, concluindo o processo, resultando no exemplo da Figura 20.

Figura 20. Exemplo de resultado final de detecção de falhas em linhas de plantio. Nesse exemplo específico foram encontrados 3.487 pixels de falhas, que com uma resolução espacial de 5cm/pixel, resultam em 174,35 metros de falhas.



Fonte: Os autores

4. RESULTADOS

Os resultados foram satisfatórios nos *datasets* escolhidos, porém, a acurácia depende intimamente da qualidade da imagem resultante da predição da CNN. Dessa forma, deduz-se que no máximo serão obtidos 92% acertos, devido ao F1 Score atingido no treinamento da rede ter sido aproximadamente 92%.

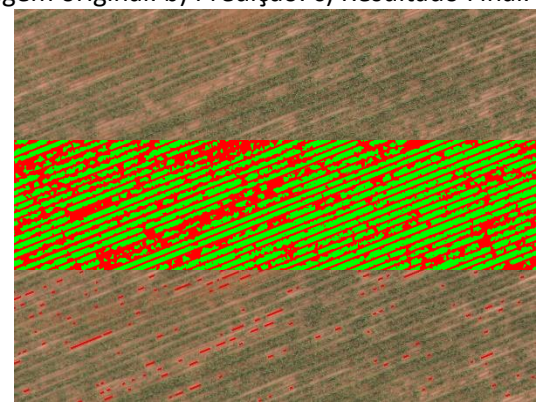
Também é importante salientar que o método proposto é bom para detectar falhas em linhas de plantio paralelas e sem curvas, já que o método depende da Transformada de Hough para Linhas para desenhar as linhas, destacar e calcular as falhas. Um exemplo de imagem processada por esse método pode ser observado na Figura 21.

Figura 21. Exemplo de resultado obtido. a) Imagem original. b) Predição. c) Resultado Final.

a)

b)

c)



Fonte: Os autores.

5. CONCLUSÕES

Este trabalho teve como objetivo propor um método para detecção de falhas em linhas de plantio em imagens obtidas por VANT, utilizando uma rede neural convolucional para fazer a classificação, separando a cultura de interesse do solo e das ervas daninhas, e posteriormente através da aplicação de operadores morfológicos e processamento digital de imagens chegar a um resultado satisfatório.

Concluiu-se que o método utilizado teve um bom desempenho nos *datasets* escolhidos, porém, os resultados são intimamente ligados à qualidade da predição feita pela CNN, que obteve um F1 Score de aproximadamente 92%. Vale salientar também que o método proposto funciona bem para linhas de plantio paralelas e sem curvas, já que depende da Transformada de Hough para Linhas para desenhar as linhas destacar e calcular as falhas.

Seria interessante desenvolver um método para casos em que as linhas de plantio não sejam somente retas, por exemplo, combinando Transformada de Hough para Linhas com a Transformada de Hough para Elipses ou Círculos.

REFERÊNCIAS

ALVES, M. O.; FERREIRA, R. V.; GALLIS, R. B. A. Otimização da Identificação de Falhas de Plantio na Cana-de-Açúcar com Uso de Geoprocessamento. *In*: CONGRESSO BRASILEIRO DE AGROINFORMÁTICA, 10., 2015, Ponta Grossa, Pr. **Anais** [...]. Ponta Grossa, Pr: UEPG - PPGCA, 2015. p. 268-278.

BAH, M D.; HAFIANE, A.; CANALS, R. CRowNet: deep network for crop row detection in UAV images. **IEEE Access**. *[S. l.]*, p. 5189-5200. 19 dez. 2019. https://doi.org/10.1109/ACCESS.2019.2960873

CANAONLINE. **Usinas e produtores não podem se conformar com um canavial cheio de falhas**. 2016. Disponível em: http://www.canaonline.com.br/conteudo/usinas-e-produtores-nao-podem-se-conformar-com-um-canavial-cheio-de-falhas.html. Acesso em: 08 dez. 2020.

CRULHAS, J. P. R.; ARTERO, A. O.; PITERI, M. A.; SILVA, F. A.; PEREIRA, D. R.; ELER, D. M.; PAPA, J. P.; ALBUQUERQUE, V. H. C. Blank Spots Identification on Plantations. **IEEE Latin America Transactions**. *[S. l.],* p. 2115-2121. ago. 2018. https://doi.org/10.1109/TLA.2018.8528224

FAO. **The future of food and agriculture**: trends and challenges. Roma: Food and Agriculture Organization of the United Nations, 2017. 163 p.

HE, K.; ZHANG, X.; REN, S.; SUN, J. **Deep Residual Learning for Image Recognition**. 2015. Disponível em: https://arxiv.org/abs/1512.03385. Acesso em: 23 nov. 2021.

HENRIQUE, A. Tecnologia Agrícola: drones para análise de falhas dos canaviais. Drones para análise de falhas dos canaviais. **RPAnews - cana & indústria**, ed. 199, - jun., 2018. Disponível em: https://revistarpanews.com.br/tecnologia-agricola-drones-para-analise-de-falhas-dos-canaviais. Acesso em: 15 set. 2020.

HOUGH Line Transform: OpenCV-Python Tutorials beta documentation. OpenCV-Python Tutorials beta documentation. 2016. Disponível em: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html. Acesso em: 02 nov. 2021.

HOUGH, P. V. C. **Method and means for recognizing complex patterns**, U.S. Patent 3,069,654, Dec. 18, 1962.

KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. A review of the use of convolutional neural networks in agriculture. **The Journal of Agricultural Science**. *[S. l.]*, v. 156, n. 3, p. 1-11. 24, maio 2018. https://doi.org/10.1017/S0021859618000436

LONG, J.; SHELHAMER, E.; DARRELL, T. **Fully convolutional networks for semantic segmentation**. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3431-3440. https://doi.org/10.1109/CVPR.2015.7298965

MONTEIRO, A. A. O.; VON WANGENHEIM, A. **Orthomosaic Dataset of RGB aerial Images for Weed Mapping. 2019.** INCoD Datasets Repository LAPIX/UFSC. Disponível em: http://www.lapix.ufsc.br/weed-mapping-sugar-cane. Acesso em: 4 dez. 2020.

NUMPY. 2021. Disponível em: https://numpy.org/. Acesso em: 08 nov. 2021.

OPENCV. 2021. Disponível em: https://opencv.org/. Acesso em: 08 nov. 2021.

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. https://doi.org/10.1109/TSMC.1979.4310076

PEREIRA JUNIOR, P. C.; VON WANGENHEIM, A. **Orthomosaic Dataset of RGB aerial Images for Crop Rows Detection.** 2019. INCoD Datasets Repository LAPIX/UFSC. Disponível em: http://www.lapix.ufsc.br/crop-rows-sugar-cane. Acesso em: 4 dez. 2020.

PIERCE, F. J.; NOWAK, P. ASPECTS OF PRECISION AGRICULTURE. *In*: DONALD L. SPARKS (EUA) (ed.). **Advances in Agronomy**. *[S. l.]*: Academic Press, 1999. p. 1-85. https://doi.org/10.1016/S0065-2113(08)60513-1

SAATH, K. C. O.; FACHINELLO, A. L. Crescimento da demanda mundial de alimentos e restrições do fator terra no Brasil. **Revista de Economia e Sociologia Rural**, [S.l.], v. 56, n. 2, p. 195-212, jun. 2018. Trimestral. FapUNIFESP (SciELO). http://dx.doi.org/10.1590/1234-56781806-94790560201.

SILVA, P. R. A. Plantio sem erros. **Revista Cultivar Máquinas**, Pelotas, v. 1, n. 152, p. 30-31, jun. 2015. Disponível em: https://www.grupocultivar.com.br/revistas/27. Acesso em: 13 set. 2020.

STOLF, R. Metodologia de avaliação de falhas nas linhas de cana-de-açúcar. **Revista STAB**, Piracicaba, v. 4, n. 6, p. 22-36, 1986.
STOLF, R; GARCIA, T. B.; NERIS, L. O.; TRINDADE JUNIOR, O.; REICHARDT, K. Avaliação de falhas em cana de açúcar segundo o método de Stolf utilizando imagens aéreas de alta precisão obtidas por VANT. Revista **STAB**, v. 34, n. 4, p. 32-39, mar./abr., 2016.

TENSORFLOW. 2021. Disponível em: https://www.tensorflow.org/?hl=pt-br. Acesso em: 08 nov. 2021.

WERNER, V. **Análise econômica e experiência comparativa entre agricultura de precisão e tradicional**. 2007. 133 f. Tese (Doutorado) - Curso de Engenharia Agrícola, Centro de Ciências Rurais, Universidade Federal de Santa Maria, Santa Maria, RS, 2007.

YU, F.; KOLTUN, V. **Multi-Scale Context Aggregation by Dilated Convolutions**. 2016. Disponível em: https://arxiv.org/abs/1511.07122v3. Acesso em: 26 nov. 2021.

ZHANG. T. Y., SUEN. C. Y. A Fast Parallel Algorithm for Thinning Digital Patterns. **ACM**, v. 27, n. 3, 1984. https://doi.org/10.1145/357994.358023

ZHAO, H.; SHI, J.; QI, X.; WANG, X.; JIA, J. **Pyramid Scene Parsing Network**. 2017. Disponível em: https://arxiv.org/abs/1612.01105v2. Acesso em: 28 out. 2021.

ZHOU, B.; ZHAO, H.; PUIG, X.; FIDLER, S.; BARRIUSO, A.; TORRALBA, A. **Scene Parsing through ADE20K Dataset**. Computer Vision and Pattern Recognition (CVPR), 2017. https://doi.org/10.1109/CVPR.2017.544

ZHOU, B.; ZHAO, H.; PUIG, X.; XIAO, T.; FIDLER, S.; BARRIUSO, A.; TORRALBA, A. Semantic Understanding of Scenes through the ADE20K Dataset. **International Journal of Computer Vision (IJCV)**, 2016. Disponível em: https://arxiv.org/pdf/1608.05442.pdf. Acesso em: 14 dez. 2021.