

IDENTIFICAÇÃO DE FALHAS DE PLANTIO EM LAVOURAS DE CANA-DE-AÇÚCAR

João Pedro Nascimento de Lima, Embrapa Agricultura Digital, lima.jp98@hotmail.com

Eduardo Antônio Speranza, Embrapa Agricultura Digital, eduardo.speranza@embrapa.br

Luiz Antonio Falaguasta Barbosa, Embrapa Agricultura Digital, luiz.barbosa@embrapa.br

Geraldo Magela de Almeida Cançado, Embrapa Agricultura Digital, geraldo.cancado@embrapa.br

Inácio Henrique Yano, Embrapa Agricultura Digital, inacio.yano@embrapa.br

RESUMO. A PRESENÇA DE FALHAS DE PLANTIO ESTÁ ENTRE OS MAIORES MOTIVOS PARA QUEBRA DE PRODUÇÃO DA CANA-DE-AÇÚCAR, SENDO QUE EM ALGUNS CASOS É NECESSÁRIO PROCEDER-SE O REPLANTIO DAS ÁREAS AFETADAS. SENDO IMPORTANTE, PORTANTO, IDENTIFICAR TAIS ÁREAS, TANTO PARA O REPLANTIO, COMO TAMBÉM, PARA ESTIMATIVA DA PRODUTIVIDADE DA LAVOURA. AS FALHAS PODEM SER OCASIONADAS POR PISOTEIOS NAS LINHAS DE CULTIVO, POR MAQUINÁRIOS DESCALIBRADOS DURANTE AS OPERAÇÕES DE COLHEITA, ENTRE OUTROS FATORES. CONSIDERA-SE COMO FALHAS DE PLANTIO, NO CASO ESPECÍFICO DA CANA-DE-AÇÚCAR, ESPAÇOS ENTRE PLANTAS MAIORES QUE 50 CM, PORQUE A PARTIR DESSE TAMANHO PASSARIAM A INTERFERIR NA PRODUTIVIDADE DA LAVOURA. COMO AS LAVOURAS DE CANA-DE-AÇÚCAR PERMITEM VÁRIOS CORTES, ANTES QUE SE FAÇA A REFORMA DO CANAVIAL, MOMENTO EM QUE SE FAZ NOVO PLANTIO, AS FALHAS ACABAM POR AFETAR A PRODUTIVIDADE DA CANA-DE-AÇÚCAR POR VÁRIOS ANOS, PODENDO, POR ISSO, CAUSAR GRANDES PREJUÍZOS ECONÔMICOS. ASSIM EXISTEM DIVERSOS TRABALHOS SENDO DESENVOLVIDOS PARA A IDENTIFICAÇÃO E MAPEAMENTO DAS FALHAS POR MEIO DE ANÁLISES DE IMAGENS DE AERONAVES REMOTAMENTE PILOTADAS (ARPs), COM O INTUITO DE SE EVITAR O PROCESSO CUSTOSO E DEMORADO DE PROCEDIMENTOS MANUAIS. DENTRE OS MÉTODOS PARA IDENTIFICAÇÃO DE OBJETOS POR PROCESSAMENTO DE IMAGENS ESTÃO AS TÉCNICAS DE APRENDIZADO DE MÁQUINA. ESTE TRABALHO TEM COMO OBJETIVO A IDENTIFICAÇÃO DE FALHAS DE PLANTIO EM LAVOURAS DE CANA-DE-AÇÚCAR EM IMAGENS CAPTURADAS POR ARPs E FOI REALIZADO COM A REDE NEURAL YOLOV5, QUE É UMA REDE DE APRENDIZADO PROFUNDO BASTANTE DIFUNDA E DE FÁCIL UTILIZAÇÃO.

Palavras-chave. *Aprendizado de máquina, Lavouras de cana-de-açúcar, Processamento de imagens, YOLOv5*

ABSTRACT. THE PRESENCE OF PLANTING FAILURES IS AMONG THE BIGGEST REASONS FOR THE DROP IN SUGARCANE PRODUCTION, AND IN SOME CASES, IT IS NECESSARY TO REPLANT THE AFFECTED AREAS. THEREFORE, IT IS VALUABLE TO IDENTIFY SUCH AREAS. TO AID IN REPLANTING AND ESTIMATING CROP PRODUCTIVITY REASONS. FAILURES CAN BE CAUSED BY TRAMPLING ON CROP LINES AND UNCALIBRATED MACHINERY DURING HARVESTING OPERATIONS, AMONG OTHER FACTORS. TO BE CONSIDERED SUGARCANE PLANTING FAILURE, THE SPACES BETWEEN PLANTS SHOULD BE LARGER THAN 50 CM. BECAUSE OF THAT SIZE, THEY WOULD INTERFERE WITH CROP PRODUCTIVITY. AS SUGARCANE IS A SEMI-PERENNIAL CROP, THE FAILURES AFFECT THE SUGARCANE PRODUCTIVITY FOR SEVERAL YEARS AND MAY CAUSE SEVERAL ECONOMIC DAMAGES. THUS, SEVERAL WORKS WERE DEVELOPED FOR PLANTING FAILURE IDENTIFICATION USING REMOTE PILOTED AIRCRAFT (RPA) IMAGERY TO AVOID COSTLY AND TIME-CONSUMING MANUAL PROCEDURES. AMONG THE METHODS FOR IDENTIFYING OBJECTS BY IMAGE PROCESSING ARE MACHINE LEARNING TECHNIQUES. THIS WORK AIMS TO IDENTIFY PLANTING FAILURES IN SUGARCANE CROPS IN IMAGES CAPTURED BY RPAs. WE USED THE YOLOV5 NEURAL NETWORK. THIS NETWORK IS A WIDESPREAD AND EASY-TO-USE DEEP LEARNING NETWORK.

Keywords. *Image processing, Machine learning, Sugarcane crops, YOLOv5*

1. INTRODUÇÃO

O Brasil destaca-se como maior produtor mundial de açúcar e segundo maior produtor de etanol em virtude de sua produção de cana-de-açúcar, que é cultivada em praticamente todo o território nacional, especialmente no estado São Paulo, que é o maior estado produtor (NOCELLI, 2017). A indústria sucroenergética também é importante na cogeração de energia elétrica, que contribui para a sustentabilidade das usinas e cujo excedente é vendido para concessionárias de distribuição de eletricidade, além de uma série de outros produtos como melaço, aguardente, etc.

Apesar de sua grande importância econômica, a rentabilidade da produção sucroenergética depende de constantes aprimoramentos nos processos produtivos em busca da redução de custos e aumento da produtividade (YANO et al., 2021). Em função disso, diversos estudos têm sido realizados no sentido de aumento de produtividade da cana-de-açúcar (GASPAROTTO, 2020).

Dentre os problemas que causam perda da produtividade da cana-de-açúcar estão as falhas de plantio, que são causadas, entre outros fatores, devido a falta de gemas viáveis por colmo, danos nas gemas presentes nos toletes na operação de plantio, distância errada entre os sulcos, umidade e adubação errada, pisoteio nas linhas de cultivo e maquinário descalibrado em operações de colheita (BARROS e MILAN, 2010). A presença de falhas de plantio afeta a produtividade, justificando sua identificação para estimativas de quebra de produção e em alguns casos para operações de replantio, utilizando principalmente mudas pré-brotadas (MPB). Consideram-se falhas, espaços vazios medidos de centro a centro dos colmos ao nível do solo e maiores que 50 cm (LEMES et al., 2021).

A identificação e o mapeamento manual de falhas de plantio é um processo custoso e demorado. Por isso, existem diversos trabalhos no sentido de proceder a identificação de falhas por meio de análise de imagens capturadas a partir de ARPs (Aeronaves Remotamente Pilotadas) (BARBOSA, 2021, D SILVA, 2020). Os ARPs, apesar da pouca autonomia de voo, conseguem percorrer grandes áreas em um curto espaço de tempo e têm resolução espacial suficiente para proceder-se a identificação de falhas.

Há trabalhos que procuram identificar falhas depois de traçar as linhas de cultivo na lavoura (D SILVA, 2020), mas limitam-se a apresentar bons resultados somente quando a lavoura possui linhas de cultivo retas ou de curvas suaves. O objetivo deste trabalho é identificar falhas de plantio, mesmo quando a lavoura apresenta curvas. Por isso, a solução escolhida foi a utilização de ferramentas de aprendizado profundo.

Neste trabalho foi utilizada a rede neural de aprendizado profundo YOLOv5 (2022), que foi escolhida devido a sua facilidade de uso, pela grande quantidade de tutoriais disponíveis na Internet e por já estar bastante consolidada, estando atualmente na sua quinta versão (JIANG et al., 2022).

2. FUNDAMENTAÇÃO TEÓRICA

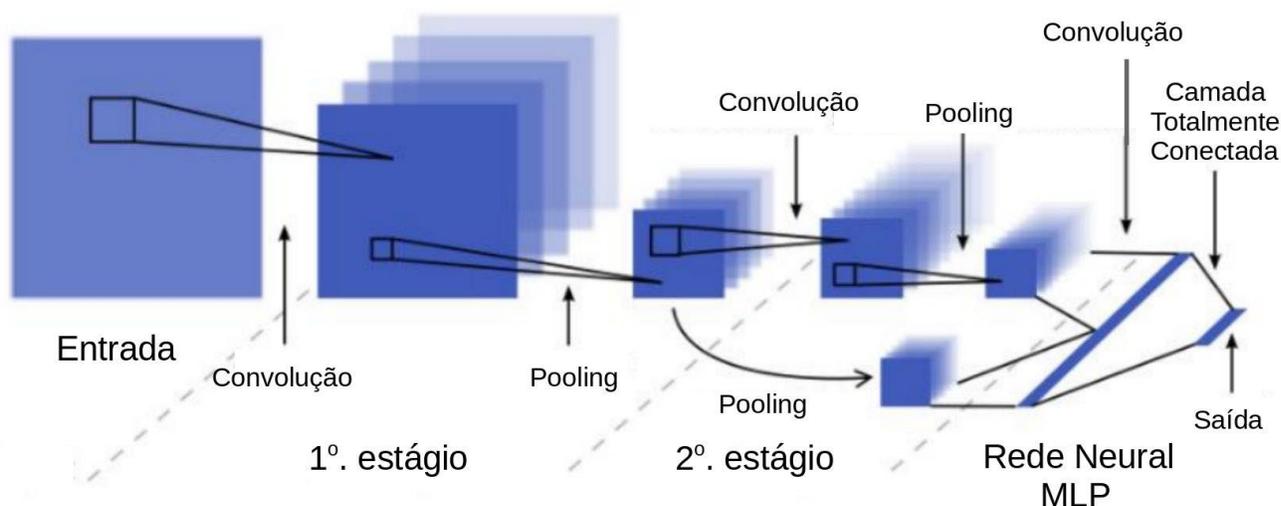
2.1 REDES NEURAS CONVOLUCIONAIS

Redes Neurais Convolucionais (RNC) têm apresentado excelentes resultados no processo de reconhecimento de padrões em imagens, sendo utilizadas para uma grande quantidade de aplicações: na agricultura como detecção de plantas daninhas em lavouras de soja (FERREIRA, 2017); na área da saúde como diagnóstico de leucemia (VOGADO et al., 2019); na astronomia para detecção de

supernovas (MONTES et al., 2021), apenas para citar alguns exemplos.

Assim como as primeiras redes neurais, as RNC são formadas por várias camadas, algumas delas intermediárias para extração de padrões em mapas de características e uma camada totalmente conectada no final, conforme Figura 1. Nas camadas intermediárias são realizadas as operações de convolução e *pooling* para extrair as características e reduzir as dimensões de entrada. Após várias operações de convolução e *pooling*, representadas na Figura 1 pelos estágios 1 e 2, o mapa de características será submetido à camada totalmente conectada, por exemplo, uma rede neural multilayer perceptron (MLP).

Figura 1 – Arquitetura de uma Rede Neural Convolutacional

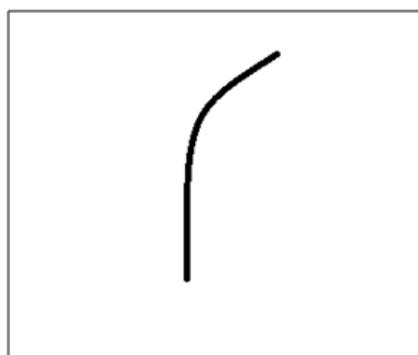


Fonte: Adaptado de DEWA et al. (2018)

Na operação de convolução, áreas específicas de maior significância da matriz de duas dimensões da entrada são selecionadas (Figura 2) para criação de filtros, cujos valores iniciais são aleatórios e, posteriormente, são ajustados no processo de treinamento da rede. No processo de reconhecimento, o filtro percorre a imagem, em um processo de multiplicação, que em caso de compatibilidade gerará valores altos, caso contrário, serão gerados valores próximos de zero. A Figura 3a mostra uma subimagem, também conhecida como *receptive field*, onde será aplicado o filtro, conforme Figura 3b e Equação 1 (ALVES, 2018).

Figura 2 – Filtro

0	0	0	0	0	30	0
0	0	0	0	30	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	30	0	0	0
0	0	0	0	0	0	0

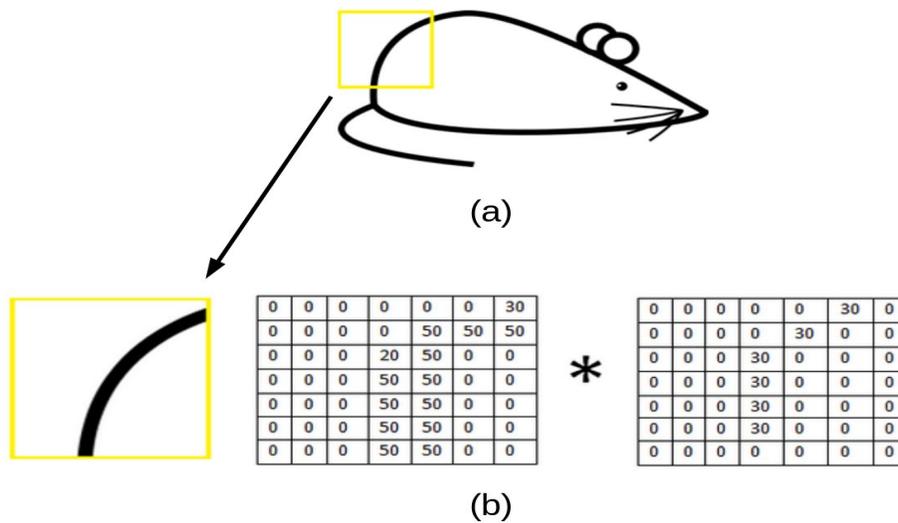


Exemplo de

nbro de 2022

Fonte: ALVES (2018)

Figura 3 – (a) *Receptive Field* (b) Aplicação de Filtro em *Receptive Field*



Fonte: adaptado de ALVES (2018)

$$(1) \quad (1) \quad Q_j = f \left(\sum_{i=1}^N I_{i,i} * K_{i,j} + B_j \right),$$

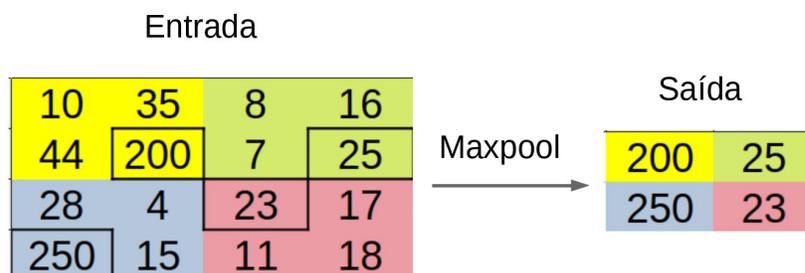
Q_j = Matriz de saída da operação de convolução;

f = função de ativação aplicada à somatória da multiplicação do *receptive field* (matriz $I_{i,j}$) pelo filtro (matriz $K_{i,j}$) somado ao bias (B_j);

A Equação 1 apresenta a saída da operação de convolução (Q_j), que é o resultado da somatória da multiplicação do *receptive field* (matriz $I_{i,j}$) pelo filtro (matriz $K_{i,j}$) somado ao bias (B_j), depois de aplicada a função de ativação. Existem várias funções de ativação, usadas para melhorar o desempenho do processo de treinamento das redes neurais, tais como função sigmoide e tangente hiperbólica, entre outras, mas a que tem apresentado melhores resultados para redes convolucionais é a ReLU. Uma vez aplicada, essa função mantém os valores positivos e zera os valores negativos (DEWA et al.,2018; Alves, 2018).

A operação de *pooling* serve para reduzir ainda mais a saída da camada anterior. Essa operação utiliza uma unidade de área pequena, por exemplo, uma matriz 2x2 para percorrer a matriz da operação anterior e selecionar um único valor para representar essa área, assim uma matriz de entrada 4x4 é reduzida para o tamanho de uma matriz 2x2. O método mais utilizado para essa operação de *pooling* é o *maxpooling*, no qual apenas o maior valor é selecionado para representar a pequena área, conforme Figura 4 (Alves, 2018).

Figura 4 – Exemplo de Maxpool



Fonte: adaptado de ALVES (2018)

3. MATERIAIS E MÉTODOS

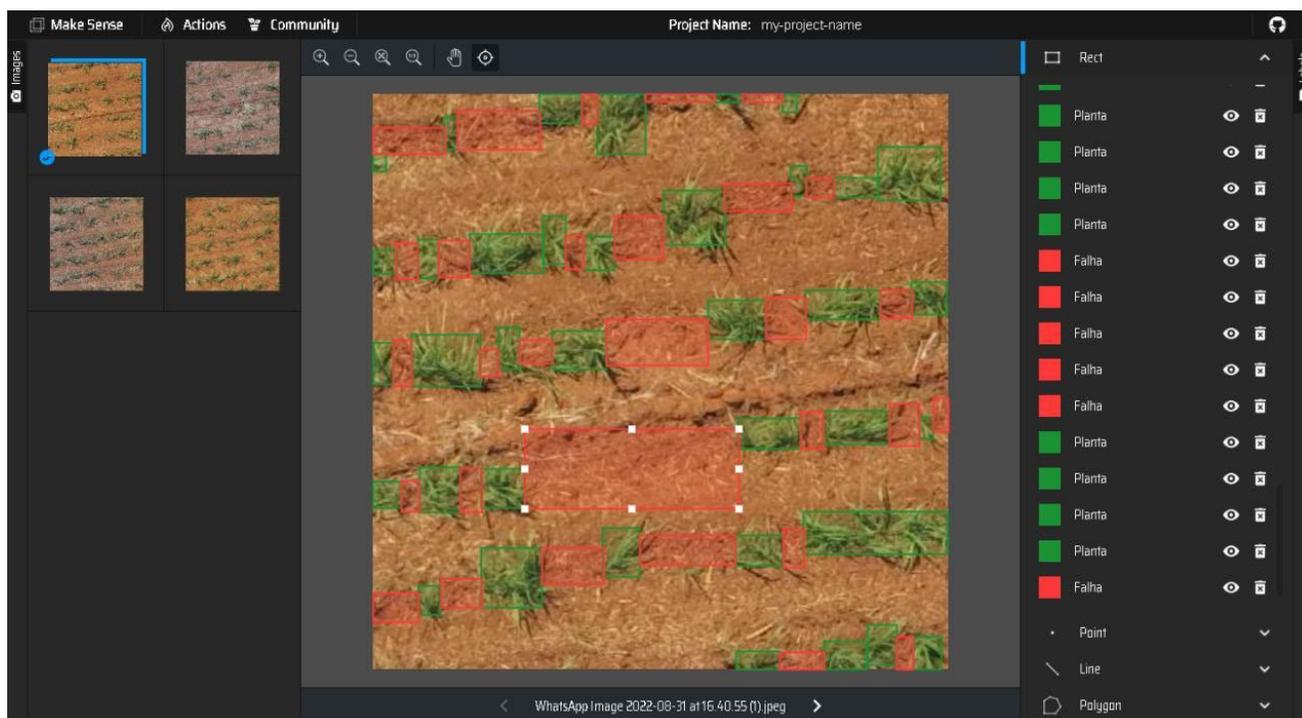
O processo de identificação de falhas foi dividido em quatro fases: recorte das imagens para treinamento e validação, marcação dos objetos a serem identificados, treinamento da rede e teste. O recorte foi feito utilizando-se a ferramenta GIMP 2.10 (GIMP, 2022), com recortes de subimagens de 400x400 pixels de um mosaico de uma lavoura de cana-de-açúcar, cedido pelo projeto Embrapa-Coplacana. Esse mosaico foi construído com imagens capturadas/obtidas por um ARP de uma lavoura

da região de Piracicaba/SP. Os recortes são necessários para se ter imagens para treinamento e validação da rede neural convolucional todas do mesmo tamanho.

Uma vez tendo as imagens de treinamento e validação prontas, o próximo passo é a marcação dos objetos de interesse. Neste caso, foram marcados duas classes de objetos: a primeira classe ou classe 0 foi constituída de falhas de plantio e a segunda classe ou classe 1 foi constituída de plantas, neste caso, de cana-de-açúcar. A ferramenta utilizada para este procedimento foi a plataforma online de código aberto MAKESENSE (2022), licenciada pela GPLv3.

O processo de marcação consiste em fazer upload das imagens na plataforma MAKESENSE, em seguida, pode-se criar os rótulos (*labels*) para as classes que se deseja marcar nas imagens. O próximo passo é abrir as imagens e para cada imagem submetida deve-se contornar com retângulos as classes a serem identificadas; neste caso, falhas de plantio ou plantas, conforme Figura 5. Ao concluir o processo é só fazer o download dos rótulos, tanto das imagens usadas para treinamento quanto de validação.

Figura 5 – Imagem da plataforma MAKESENSE com labels para falhas de plantio marcadas em vermelho e plantas marcadas em verde

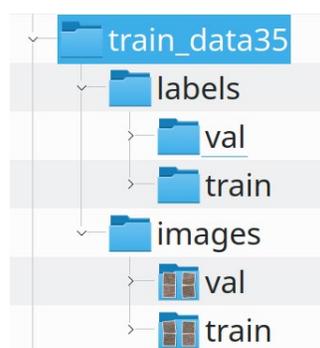


Fonte: próprios autores

Tendo as imagens e os respectivos rótulos, deve-se criar um diretório a ser usado no processo de treinamento do classificador. Neste caso, conforme descrito anteriormente, foi escolhido o

classificador YOLOv5. Esse diretório deverá ter dois subdiretórios: um com imagens e outro com rótulos. Estes por sua vez devem ter cada um dois outros subdiretórios: um para treinamento e outro para validação, conforme estrutura da Figura 6.

Figura 6 – Estrutura de Diretórios para Treinamento do YOLOv5



Fonte: próprios autores

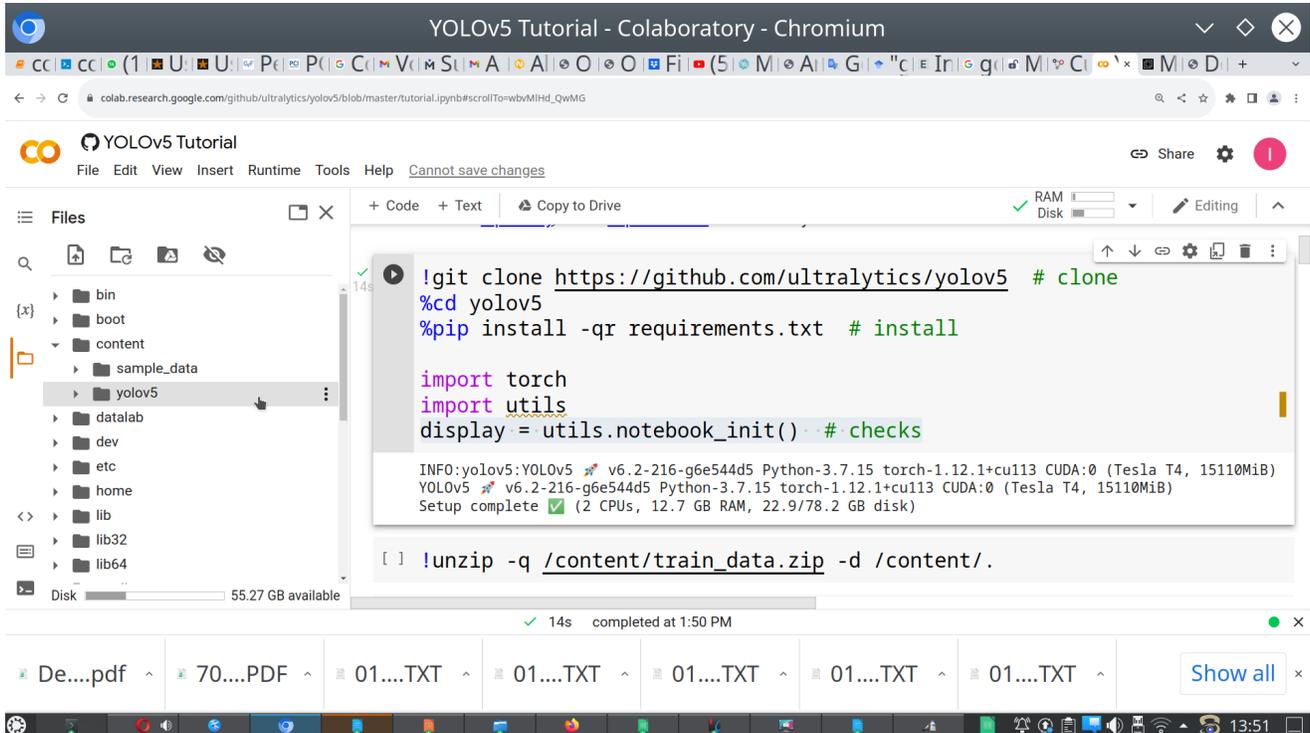
O passo seguinte é o treinamento da rede neural convolucional YOLOv5. No entanto, devido a um melhor desempenho utilizando-se máquinas com GPU, optou-se por realizar o processamento no Google Colab (ALNAJJAR, 2021; Google Colab, 2022), que fornece ambiente próprio para esta necessidade de processamento gratuitamente de forma temporária. No Google Colab, é possível baixar os programas do YOLOv5 diretamente do repositório GitHub (SHARMA et al., 2017) e criar o ambiente para treinamento e geração de modelos do YOLOv5, conforme lista de comandos da Tabela 1. A Figura 7 apresenta o ambiente do Google Colab pronto para treinamento do YOLOv5, cuja estrutura de diretórios fica abaixo do diretório “/content”.

Tabela 1: Comandos de Setup YOLOv5 no Google Colab

Linha	Comando
1	<code>!git clone https://github.com/ultralytics/yolov5 # clone</code>
2	<code>%cd yolov5</code>
3	<code>%pip install -qr requirements.txt # install</code>
4	<code>import torch</code>
5	<code>import utils</code>
6	<code>display = utils.notebook_init() # checks</code>

Fonte : próprios autores

Figura 7 – Ambiente no Google Colab pronto para treinamento do YOLOv5



Fonte : próprios autores

Com o Setup do YOLOv5 pronto, deve-se compactar o diretório de treinamento e validação apresentado na Figura 6. Em seguida, fazer o upload na estrutura de diretórios do Google Colab, por exemplo em “/content”. Também é necessário criar um arquivo de configuração com as informações sobre o caminho das imagens e labels para treinamento e validação, bem como, as classes dos objetos a serem identificados, conforme conteúdo apresentado na Tabela 2, e fazer o upload do mesmo; neste exemplo, o upload foi para o diretório “/content/YOLOv5/data”. Observando o conteúdo do arquivo falha.yaml, é possível ver as duas classes de objetos a serem identificados, sendo a classe “0: f” para identificar falhas de plantio e a classe “1: c” para plantas ou cana-de-açúcar.

Tabela 2: Conteúdo do Arquivo de Configuração (falha.yaml) para Treinamento do YOLOv5

Linha	Comando
-------	---------

```
1 path: ../. # dataset root dir
2 train: ./train_data35/images/train # train images (relative to 'path') 35 images
3 val: ./train_data35/images/val # val images (relative to 'path') 13 images
4 test: # test images (optional)
5
6 # Classes
7 names:
8 0: f
9 1: c
10
11 # Download script/URL (optional)
12 download: https://ultralytics.com/assets/coco128.zip
13
```

Fonte : próprios autores

Para iniciar o treino, é necessário antes descompactar o diretório com os dados de treinamento (train_data35) com o comando: “!unzip -q /content/train_data.zip -d /content/.”. Posteriormente, pode-se executar para início do treinamento o comando: “!python /content/yolov5/train.py --img 400 --batch 16 --epochs 300 --data falha.yaml --weights yolov5s.pt --cache”. Finalizado o treino, pode-se fazer o upload de novas imagens para constatar a eficiência no processo de reconhecimento de falhas de plantio. Para realização dos testes pode-se executar o comando: “!python /content/yolov5/detect.py --weights /content/yolov5/runs/train/exp3/weights/best.pt --img 400 --conf 0.25 --source /content/train_data/test2.jpeg”.

4. RESULTADOS E DISCUSSÃO

O resultado deste trabalho são as imagens com as falhas e plantas identificadas. A Figura 8a apresenta a imagem original e a Figura 8b apresenta a imagem com os objetos identificados, sendo a letra “f” atribuída para falhas de plantio e a letra “c” para plantas ou cana-de-açúcar. Para uma melhor visualização, foram impressas na Figura 9 apenas as identificações de falhas de plantio que são objeto deste estudo.

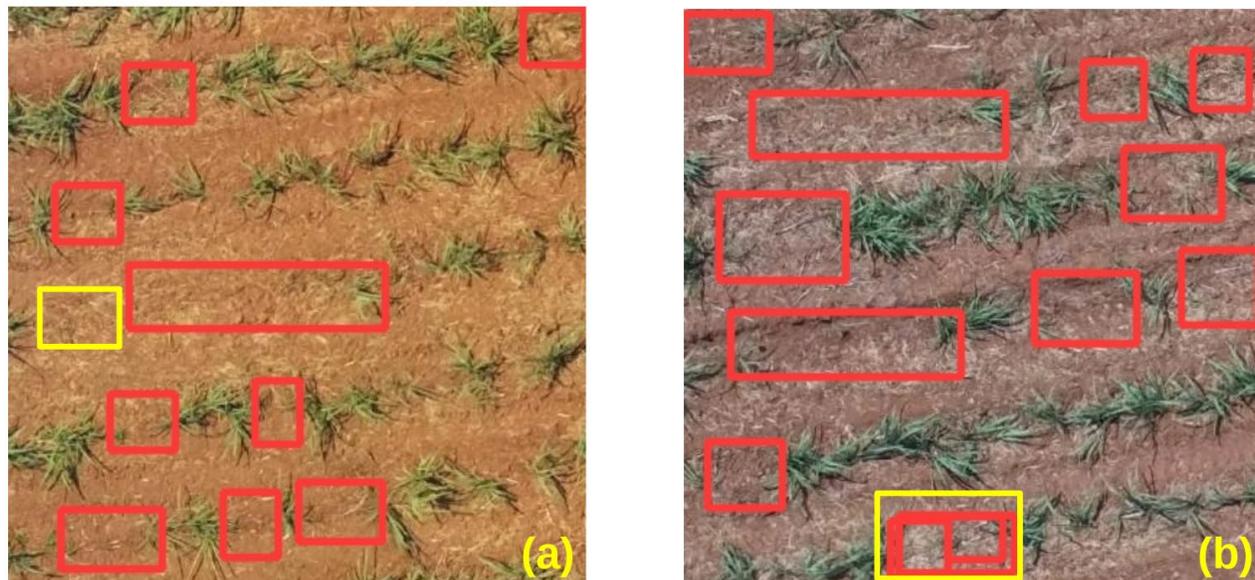
A Figura 9 apresenta com clareza a identificação das falhas e na mesma foram selecionados dois problemas encontrados e demarcados com um retângulo amarelo. O primeiro problema na Figura 9a foi a de marcação de falha com área inferior à área total da falha e na Figura 9b houve sobreposição e marcação em duplicidade da falha de plantio. No primeiro caso é que realmente pode ser considerado um problema, porque, com área demarcada inferior à real, ocorre a subestimação da representatividade da falha na lavoura. No segundo caso, como se pretende fazer um resumo das falhas encontradas, as sobreposições serão anuladas no processo de consolidação, não causando prejuízo na estimativa da presença de falhas na lavoura.

Figura 8 – (a) imagem original e (b) imagem com falhas de plantio e plantas identificadas



Fonte : próprios autore

Figura 9 – (a) Falha demarcada com área menor e (b) sobreposição e marcação em duplicidade



Fonte : próprios autore

5. CONCLUSÃO

O objetivo principal deste trabalho foi alcançado com a identificação de falhas de plantio utilizando-se a rede neural convolucional YOLOv5. Por se tratar de um trabalho inicial, ainda serão necessários ajustes para, principalmente, sanar o problema de marcação com área inferior à real, uma vez que isto constitui um problema para constatar a área total de falhas de plantio e prejudicar estimativas de safra. Conforme descrito nos resultados, existe ainda o problema da sobreposição de demarcações, mas este problema pode ser anulado sem grandes dificuldades em um processo de consolidação e mapeamento das falhas encontradas.

Este trabalho foi uma primeira tentativa de proceder-se à identificação de falhas de plantio com o uso do YOLOv5, sendo que em estudos futuros pretende-se aumentar a base de treinamento e validação com o intuito de eliminar ou reduzir as submarcações. É possível também que aumentando o número de rodadas de treinamento, esses problemas encontrados sejam reduzidos, pois os testes foram realizados com somente 300 *epochs*.

AGRADECIMENTOS

Os autores agradecem à Embrapa e à Coplacana pelas imagens cedidas para realização deste trabalho.

REFERÊNCIAS

- ALNAJJAR, Mohammad. Image-Based Detection Using Deep Learning and Google Colab. 2021.
- ALVES, Gisely. “Entendendo Redes Convolucionais (CNNs)”. [S.I.]. **MEDIUM**, 2018. Disponível em: <https://medium.com/neuronio-br/entendendo-redes-convolucionais-cnns-d10359f21184>, Acesso em: 29 out 2022.
- ALVES, Matheus Oliveira; FERREIRA, Ricardo Vicente; DE ARAÚJO GALLIS, Rodrigo Bezerra. Otimização da Identificação de Falhas de Plantio na Cana-de-Açúcar com Uso de Geoprocessamento. In: **X Congresso Brasileiro de AgroInformática, outubro de**. 2015.
- BARBOSA JÚNIOR, Marcelo Rodrigues. Mapeamento de falhas em cana-de-açúcar por imagens de veículo aéreo não tripulado. **UNESP**, 2021.
- BARROS, Fernando Ferraz; MILAN, Marcos. Qualidade operacional do plantio de cana-de-açúcar. **Bragantia**, v. 69, p. 221-229, 2010.
- D SILVA, R. R. Detection of sugarcane crop rows from uav images using semantic segmentation and random transform. 2020.
- FERREIRA, Alessandro dos Santos. Redes neurais convolucionais profundas na detecção de plantas daninhas em lavoura de soja. 2017.
- GASPAROTTO, Letícia Gonçalves. **Quantificando o efeito de algumas variáveis de manejo agrícola na produtividade da cana-de-açúcar em uma usina do Estado de São Paulo**. 2020. Tese de Doutorado. Universidade de São Paulo.

GIMP - GNU Image Manipulation Program. [S.I.]. **GIMP**, [s.d]. Disponível em: <https://www.gimp.org>, Acesso em: 05 out 2022.

GOOGLE COLAB. [S.I.]. **Google**, [s.d]. Disponível em: <https://colab.research.google.com>. Acesso em: 05 jun 2022.

JIANG, Peiyuan et al. A Review of Yolo algorithm developments. **Procedia Computer Science**, v. 199, p. 1066-1073, 2022.

LEMES, Luíla Macêdo et al. Avaliação da qualidade do plantio semi mecanizado de cana-de-açúcar com controle estatístico de processo. *Research, Society and Development*, v. 10, n. 5, p. e10710513962-e10710513962, 2021.

MAKESENE. [S.I.]. **MAKESENSE**, [s.d]. Disponível em: <https://www.makesense.ai>, Acesso em: 05 jun 2022.

MONTES, Gabriela Cardoso; LATTARI, Lucas Grassano; COELHO, Alessandra Martins. Redes Neurais Convolucionais Otimizadas para a Detecção de Supernovas. In: **Anais do XV Brazilian e-Science Workshop. SBC**, 2021. p. 1-8.

NOCELLI, Roberta Cornélio Ferreira et al. Histórico da cana-de-açúcar no Brasil: contribuições e importância econômica. **Cana-de-açúcar e seus impactos: uma visão acadêmica**, p. 13, 2017.

SHARMA, Abhishek et al. Cataloging github repositories. In: **Proceedings of the 21st International Conference on Evaluation and Assessment in Software Engineering**. 2017. p. 314-319.

VOGADO, Luis HS et al. Rede neural convolucional para o diagnóstico de leucemia. In: **Anais do XIX Simpósio Brasileiro de Computação Aplicada à Saúde**. SBC, 2019. p. 46-57.

YANO, Inácio Henrique et al. Sistema de acompanhamento de investimentos para aumento da rentabilidade e sustentabilidade do setor sucroenergético. In: **Embrapa Agricultura Digital-Artigo em anais de congresso (ALICE)**. In: ENCONTRO NACIONAL DE ENGENHARIA DE PRODUÇÃO, 41., 2021, Foz do Iguaçu. Contribuições da engenharia de produção para a gestão de operações energéticas sustentáveis: anais. Rio de Janeiro: ABEPRO, 2021., 2021.

YOLO - You only look once. [S.I.]. **YOLOv5**, [s.d]. Disponível em: <https://docs.ultralytics.com>, Acesso em: 05 jun 2022.