

DESENVOLVIMENTO DE ALGORITMOS DE PROCESSAMENTO DE IMAGENS PARA IDENTIFICAÇÃO DE PRAGAS EM LAVOURAS DE TRIGO

Otto Camargo Kuchkarian ¹; Wânderson de Oliveira Assis ²; Fernando de Almeida Martins ³; Alessandra Dutra Coelho ⁴

¹ Aluno de Iniciação Científica do Instituto Mauá de Tecnologia (IMT);

^{2,4} Professor do Instituto Mauá de Tecnologia (IMT);

³ Engenheiro do Centro de Pesquisas (CP-CEUN-IMT).

Resumo. *O presente projeto tem como objetivo desenvolver uma solução computacional para identificar pragas em plantações de trigo, utilizando algoritmos de processamento de imagens e aprendizado de máquina. A partir da seleção das pragas mais comuns nessas lavouras, foi criado um modelo utilizando a plataforma Roboflow para reconhecer a presença desses insetos. A biblioteca OpenCV foi utilizada para acessar a câmera e aplicar o modelo em tempo real. Testes extensivos foram realizados para avaliar a precisão do modelo e identificar aspectos que possam ser aprimorados em seu treinamento. Essa abordagem permitirá um monitoramento contínuo e automatizado das lavouras, facilitando a detecção precoce de pragas e a adoção de medidas corretivas, contribuindo para a produtividade e sustentabilidade da produção de trigo no Brasil.*

Introdução

A cultura do trigo é fundamental para a segurança alimentar e a economia agrícola do Brasil, sendo uma importante fonte de alimento e renda. No entanto, a produção enfrenta desafios devido à presença de pragas que comprometem a qualidade e a quantidade das colheitas, reduzindo a rentabilidade dos agricultores e demandando soluções inovadoras para mitigação (Exame Solutions, 2024).

Com o avanço de tecnologias como a Internet das Coisas (IoT) e o processamento de imagens, surgem novas possibilidades para a agricultura de precisão, como o monitoramento em tempo real e a detecção precoce de problemas nas lavouras (Exame Solutions, 2024). Estudos mostram que a combinação de processamento de imagem com *machine learning* oferece um potencial promissor para enfrentar esses desafios na agricultura (Lins et al., 2020; Cesaro Júnior et al., 2022). Combinando processamento de imagem com técnicas de *machine learning* e *deep learning* há potencial para o desenvolvimento de diversos projetos que muito contribuem para a agricultura de precisão (Barbedo, 2022). (Ferentinos, 2018). (Mohanty et al., 2016). (Zanini, 2021).

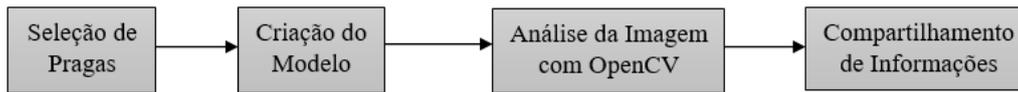
Este projeto propõe um sistema de processamento de imagem para monitoramento automatizado de plantações de trigo e identificação de pragas, utilizando Python, OpenCV e a plataforma Roboflow para criar um modelo de detecção em tempo real. A solução integrará câmeras e algoritmos de aprendizado de máquina para reconhecer pragas nas lavouras, permitindo a implementação de ações corretivas de forma ágil e eficiente.

Espera-se que essa abordagem contribua para a sustentabilidade e produtividade do trigo no Brasil, além de servir como modelo para outras culturas agrícolas. A aplicação de tecnologias avançadas na agricultura promove a eficiência e a gestão sustentável das lavouras, favorecendo o desenvolvimento do setor.

Material e Métodos

Para iniciar o projeto, foram definidas etapas que permitirão uma execução organizada e eficiente para a produção da solução computacional. As etapas estão dadas na Figura 1:

Figura 1 - Etapas do Projeto



Para o desenvolvimento do sistema de combate às pragas, foi elaborada uma lista de *softwares* e *hardwares* que serão utilizados. Estes materiais são essenciais para a criação do sistema, que tem como principal objetivo detectar as pragas e notificar o usuário sobre possíveis infestações. A seguir, estão os itens selecionados:

1. PyCharm: Ambiente de desenvolvimento integrado (IDE) utilizado para a manipulação e edição do código-fonte;
2. Python: Linguagem de programação escolhida para a implementação do projeto.
3. OpenCV (*Open Source Computer Vision Library*): Biblioteca empregada para realizar operações com a câmera, essenciais para a identificação das pragas;
4. Roboflow: Plataforma utilizada para o desenvolvimento e treinamento de modelos de visão computacional, com foco na detecção das pragas selecionadas;
5. Smtplib: Biblioteca de Python utilizada para o envio de notificações por e-mail, alertando os usuários sobre a presença de pragas;
6. Câmera: Webcam HD CAM 720p da Intelbras, utilizada para capturar imagens em tempo real das lavouras;
7. GitHub: Plataforma de desenvolvimento colaborativo utilizado para alojar o projeto na nuvem e permitir o fácil acesso para qualquer usuário, deixando o projeto *Open Source*.

Seleção de Pragas

Diversas pragas afetam as lavouras de trigo, comprometendo tanto a quantidade quanto a qualidade da colheita. Para o início do projeto, foi necessário selecionar pragas específicas que, além de serem prevalentes nas lavouras de trigo, apresentam características que facilitam a sua identificação através de técnicas de processamento de imagem.

Após uma análise cuidadosa, foi feita uma lista das espécies mais comuns em lavouras de trigo. A seguir, tem-se todas as pragas que servirão de objetos de identificação do modelo: pulgão-verde-pálido (*Metopolophium dirhodum*); piolho-da-cereja-brava (*Rhopalosiphum padi*); pulgão-verde-dos-cereais (*Schizaphis graminum*); pulgão-preto-dos-cereais (*Sipha maydis*); pulgão-da-espiga (*Sitobion avenae*) (Costa, 2023).

Com as pragas devidamente selecionadas, foram feitas buscas rigorosas em locais variados da Internet como também em banco de dados específicos da Embrapa para coletar diversas imagens de cada uma das espécies para a produção futura do modelo (Embrapa, 2024). Tendo as imagens devidamente selecionadas, a próxima etapa do projeto foi utilizando a plataforma Roboflow, criar um modelo para reconhecer cada uma das pragas escolhidas.

Criação do Modelo

A criação de um modelo de visão computacional no Roboflow é um passo crucial para o desenvolvimento do sistema de detecção de pragas em lavouras de trigo. O Roboflow é uma plataforma robusta que facilita o gerenciamento de dados visuais e a construção de modelos de *machine learning*. Para se iniciar o processo, foi necessário criar uma conta na plataforma e um projeto:

1. Registro e Criação do Projeto

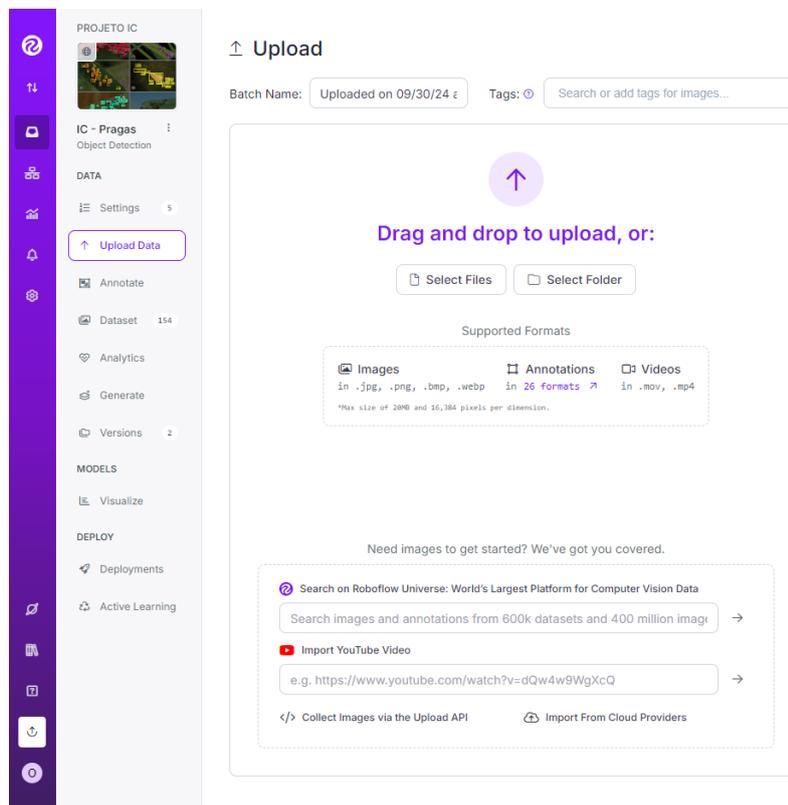
Primeiramente, acessou-se o site do Roboflow, onde foi realizada a criação de uma conta gratuita. Após o registro, fez-se o login e clicou-se na opção “*New Project*” (Novo Projeto). O projeto criado foi nomeado como “Projeto IC”.

2. Upload de Imagens

Com o projeto criado, a próxima etapa consistiu em fazer o *upload* das imagens coletadas:

- Importação: Na interface do projeto, utilizou-se a função de *upload* para adicionar as 158 imagens previamente selecionadas das pragas. O Roboflow aceita formatos comuns, como JPEG e PNG.

Figura 2 – Janela de exemplo de *Upload* do projeto

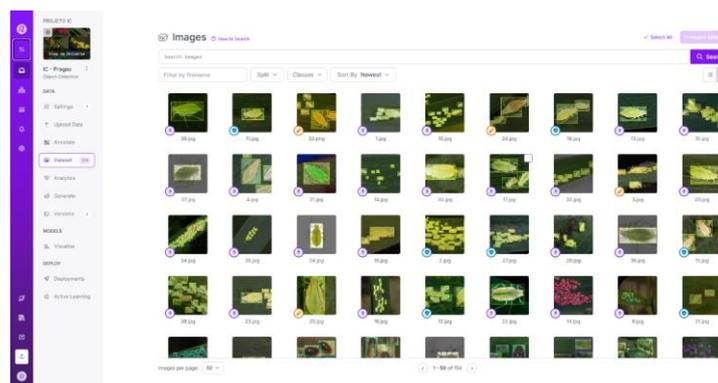


3. Anotação de Imagens

A anotação das imagens foi uma etapa crítica para o reconhecimento das pragas:

- Ferramenta de Anotação: Utilizaram-se as ferramentas de anotação disponíveis no Roboflow para desenhar caixas delimitadoras em torno das pragas visíveis nas imagens;
- Rotulagem: Cada caixa delimitadora foi rotulada com a espécie correspondente, como "*Sipha maydis*" ou "*Sitobion avenae*", sendo essa rotulagem essencial para o aprendizado supervisionado do modelo.

Figura 3 – Janela de exemplo do *Dataset* para o modelo

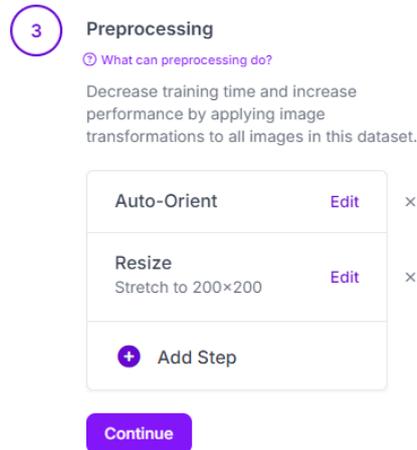


4. Pré-processamento dos Dados

Após a anotação de cada imagem, realizou-se o pré-processamento dos dados para garantir sua qualidade:

- Redimensionamento: As imagens foram ajustadas para uma dimensão uniforme, assegurando que todas tivessem a mesma proporção;
- Aumento de Dados: O Roboflow ofereceu técnicas de aumento de dados, como rotação, alteração de brilho e corte, que ajudaram a diversificar o conjunto de dados e melhorar a robustez do modelo.

Figura 4 – Janela de exemplo do Pré-processamento para o modelo

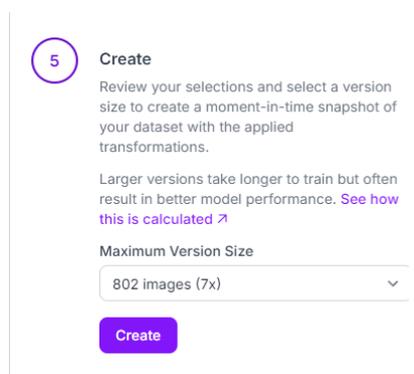


5. Treinamento do Modelo

Com a preparação dos dados, iniciou-se o treinamento do modelo:

- Seleção do Algoritmo: O Roboflow permite a escolha de diferentes algoritmos de detecção de objetos, como YOLO (*You Only Look Once*) ou SSD (*Single Shot MultiBox Detector*), devendo ser selecionado o algoritmo mais adequado para o contexto do projeto. Para o presente projeto, optou-se pelo Roboflow 3.0 *Object Detection*, devido ao seu rápido treinamento e maior precisão;
- Configuração: Configuraram-se os parâmetros de treinamento, como taxa de aprendizado, número de épocas e tamanho do lote, adaptando-os às necessidades do modelo;
- Iniciar Treinamento: Com as configurações definidas, o treinamento foi iniciado. Durante o processo, o Roboflow fornece métricas de desempenho, possibilitando a avaliação da eficácia do modelo.

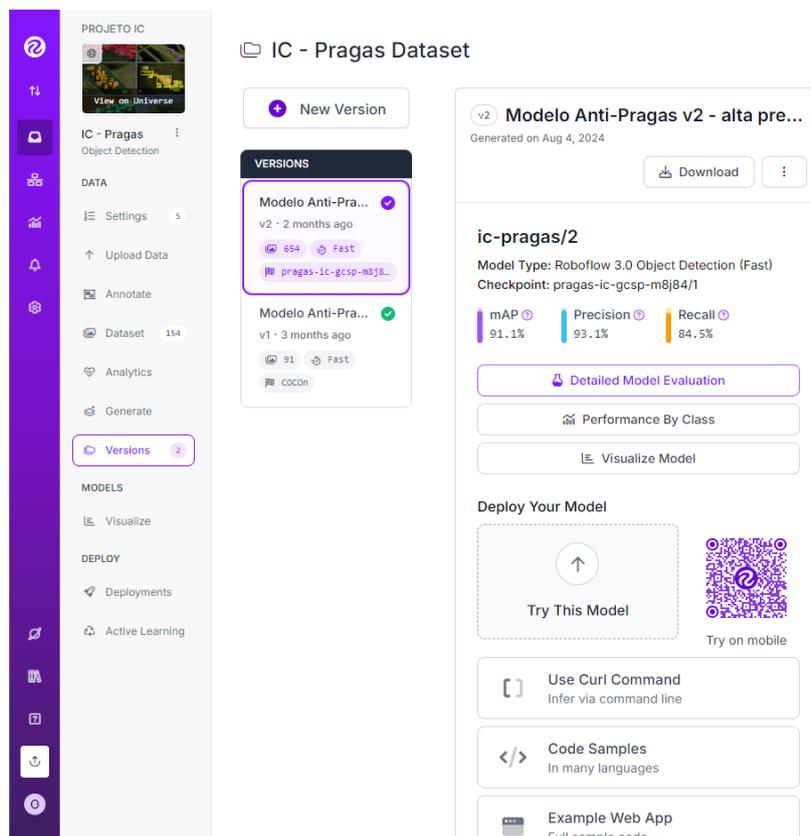
Figura 5 – Janela de exemplo da Criação do modelo



6. Finalização do Modelo

Com o treinamento feito, foram geradas duas versões do modelo, cada uma com suas vantagens e desvantagens. Na sua primeira versão, foi utilizado um *Dataset* com poucas imagens, mas com ele foi possível de identificar cada uma das espécies selecionadas sem quaisquer equívocos entre as classes de objetos. Ao contrário da segunda versão, que apresenta uma precisão muito maior da sua primeira forma, mas teve uma das espécies descartadas pelo modelo.

Figura 6 – Janela de versões do *Roboflow*



Análise da Imagem com OpenCV

Para a próxima etapa, foi necessário fazer a integração do modelo Roboflow com a linguagem Python. O sistema criado realiza, por meio de uma *WebCam*, a captura de *frames* para serem examinado pela plataforma Roboflow. Foram criados dois programas para realizar testes do sistema que utiliza a previsão de pragas por meio de uma imagem dada ou pela captura de um *frame* a cada 10 segundos com uma *WebCam* da própria máquina. Abaixo, segue o fluxograma e suas especificações:

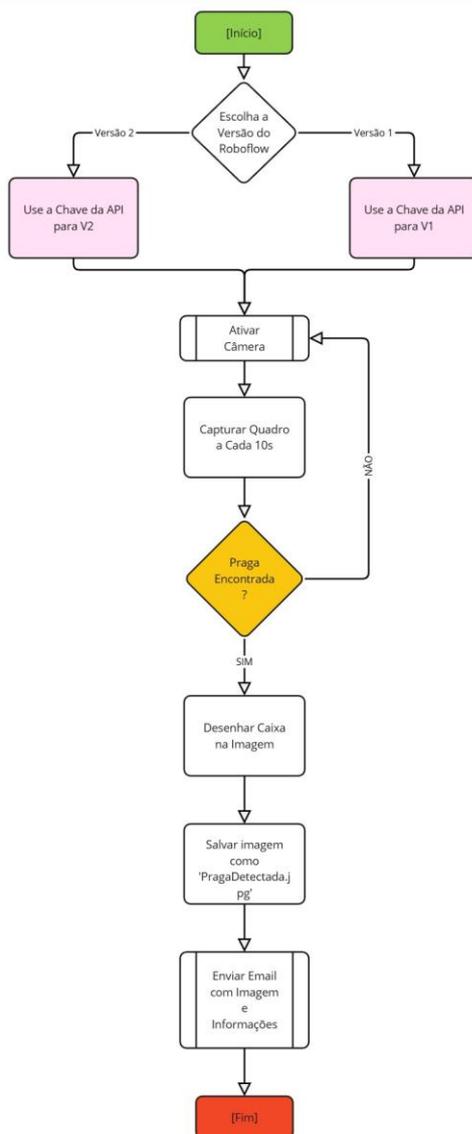
main.py

O programa se inicia perguntando ao usuário qual versão será utilizada pelo Roboflow, sendo as versões 1 e 2 disponíveis. Com isso, ele realiza uma comunicação por uma chave API para poder acessar o modelo em sua versão escolhida.

Após isso, A cada 10 segundos, o programa ativa a câmera do dispositivo em que está sendo executado e captura o quadro atual para que o modelo possa analisar a imagem em busca de pragas. Caso a imagem não possua os objetos de identificação, ela retorna à normalidade, voltando a fazer a análise quando se passar o tempo estipulado do sistema. Caso contrário, uma nova janela será exibida no computador, anunciando as pragas com quadrados de identificação, cada um contendo seu nome e grau de confiança do objeto.

Após essa exibição, o *frame* será salvo em uma imagem chamada “PragaIdentificada.jpg” e enviada para o e-mail do usuário cadastrado dentro do próprio programa, junto com outras informações, como quantidade de pragas e que horas a captura foi feita.

Figura 7 – Fluxograma para programa *mainVideo.py*



Compartilhamento de Informações

Para realizar o compartilhamento das informações, foi decidido utilizar o e-mail devido à sua fácil integração com a linguagem Python. Para implementar isso, o primeiro passo foi criar uma conta de e-mail que será responsável por enviar as informações. Além de criar a conta, foi necessário configurar medidas de segurança adequadas, como a verificação em duas etapas (*2-step verification*) e a senha de aplicativo (*app password*).

A verificação em duas etapas adiciona uma camada extra de proteção, exigindo uma segunda forma de autenticação além da senha usual, como um código enviado ao seu telefone. Esse recurso é fundamental para evitar acessos não autorizados à conta de e-mail.

Além disso, para permitir que o Python acesse e envie e-mails pelo servidor SMTP (como o Gmail, por exemplo), é necessário gerar uma "senha de aplicativo". Essa senha é específica para o aplicativo que será utilizado no *script* Python e garante que, mesmo com a verificação em duas etapas ativada, o acesso possa ser concedido ao seu código, sem comprometer a segurança da conta. Para isso, foram seguidos os seguintes passos:

1. Ativar a verificação em duas etapas: Acesse as configurações da conta de e-mail e ative o *2-step verification*. Será necessário associar um número de telefone ou outro método de segurança;
2. Gerar a senha do aplicativo: Após ativar a verificação em duas etapas, vá até a seção "Segurança" da conta e procure por "Senhas de aplicativo". Gere uma nova senha específica para o Python. Essa senha será usada no lugar da senha normal do e-mail no *script* do projeto.

Resultados e Discussão

O sistema foi testado utilizando um conjunto de imagens das cinco espécies de pragas selecionadas, a saber: pulgão-verde-pálido, piolho-da-cereja-brava, pulgão-preto-dos-cereais, pulgão-verde-dos-cereais e pulgão-da-espiga. As imagens foram obtidas de diferentes fontes para garantir uma ampla diversidade do *Dataset*.

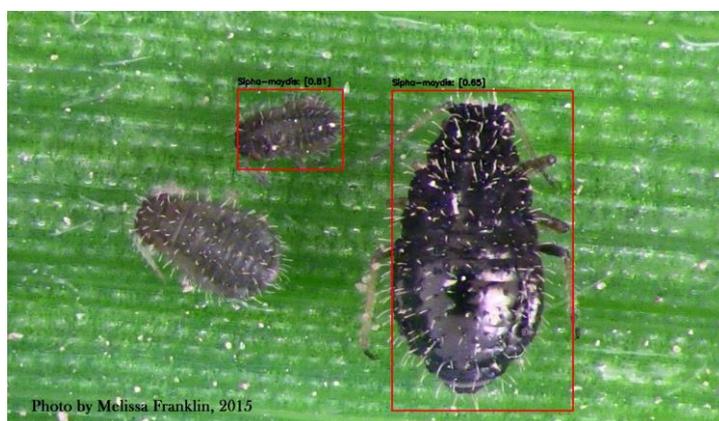
Resultado das Imagens Selecionadas

Com as imagens coletadas pela internet, foi possível treinar duas versões do modelo usando o Roboflow. A primeira versão utilizou 91 imagens para o treinamento, enquanto a segunda usou 654 imagens. A seguir estão os detalhes de cada versão.

Primeira versão

A primeira versão do modelo é capaz de identificar as 5 espécies, mas possui um grau baixo de precisão, sendo ele 84,8%, devido ao fato de que apenas 91 imagens foram utilizadas. A Figura 8 mostra um exemplo de identificação da espécie *Sipha maydis* para a primeira versão do modelo.

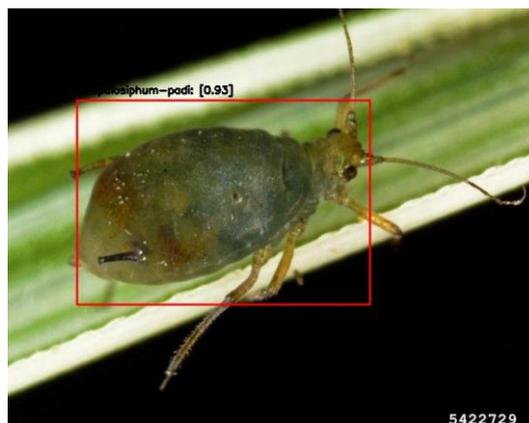
Figura 8 – Exemplo de Aplicação - Pré-processamento aplicado ao modelo 1



Segunda versão

Com um conjunto de dados expandido para 654 imagens, essa versão atingiu uma precisão significativamente maior. Entretanto, um dos desafios foi a omissão de uma das espécies (*Schizaphis graminum*) durante a classificação, o que será corrigido em futuras atualizações do modelo. A exclusão dessa espécie foi justificada pela alta semelhança com *Metopolophium dirhodum*, o que levava o modelo a confundir as duas espécies, prejudicando a acurácia das detecções. A Figura 9 mostra um exemplo de identificação da espécie *Rhopalosiphum padi* para a primeira versão do modelo.

Figura 9 – Exemplo de Aplicação - Pré-processamento e aglomerações aplicados no modelo 2



Avaliação do Modelo

A comparação entre os dois modelos gerados no Roboflow está apresentada na tabela abaixo, destacando os valores de precisão, *mAP* e *Recall*.

mAP (Mean Average Precision)

O *mAP* é uma média da precisão em diferentes pontos de *recall* (capacidade do modelo de identificar corretamente todas as instâncias de pragas presentes), o que torna essa métrica valiosa para medir a precisão global do modelo em várias classes. No contexto da detecção de pragas, o *mAP* mede a capacidade do modelo de detectar com precisão várias pragas, levando em conta o equilíbrio entre a precisão (quantos objetos detectados são realmente pragas) e o *recall* em múltiplas classes e regiões de detecção. Quanto mais próximo de 100%, melhor o modelo está em termos de acurácia geral.

Recall

O *recall* é uma métrica que indica a capacidade do modelo de identificar todos os objetos reais (pragas, no caso). Em outras palavras, ele mede a proporção de pragas que o modelo conseguiu detectar em relação ao total existente. Um *recall* alto significa que o modelo consegue localizar a maior parte das pragas presentes na imagem, mesmo que em alguns casos ele possa incluir falsos positivos. Essa métrica é essencial quando se prioriza a identificação de todas as pragas para evitar infestações não detectadas.

Tabela 1 – Comparação de Desempenho entre os Modelos *Roboflow*

Modelo	Modelo 1	Modelo 2
precisão	84,8%	93,1%
<i>mAP</i>	80,3%	91,1%
<i>Recall</i>	75,4%	84,5%

Observa-se que o modelo 2, ao excluir a espécie *Schizaphis graminum*, apresentou um desempenho superior em todas as métricas avaliadas. A precisão aumentou de 84,8% para 93,1%, o *mAP* de 80,3% para 91,1%, e o *recall* de 75,4% para 84,5%, indicando uma melhoria significativa na capacidade do modelo de detectar corretamente as pragas remanescentes.

Durante os testes com imagens reais, o sistema foi capaz de identificar as pragas corretamente na maioria dos casos, conforme ilustrado pela Figura 10. O envio das notificações por e-mail também funcionou conforme esperado, com as informações sobre a praga detectada, a quantidade identificada pelo modelo e o horário da captura sendo enviadas automaticamente ao usuário.

Figura 10 – Exemplo de e-mail que o “bot” envia



Resultados

As tabelas e gráficos gerados indicam que a segunda versão do modelo apresentou uma taxa de identificação significativamente mais alta, com menos erros de classificação, sugerindo que a quantidade de dados tem um impacto direto na robustez do modelo.

O sistema de e-mail demonstrou ser uma solução eficaz para a comunicação instantânea com os agricultores. Em todos os testes realizados, as notificações foram enviadas corretamente após a detecção de pragas, com os anexos de imagem e as informações detalhadas sobre as pragas. As informações que o e-mail envia são todas as pragas que o modelo é capaz de identificar e o número de afídeos de cada espécie encontrados no *frame* capturado (Figura 10).

Essa funcionalidade é essencial para o monitoramento contínuo das lavouras, permitindo que os agricultores tomem decisões rápidas para controlar infestações, garantindo maior produtividade e menor prejuízo. No entanto, para aprimoramento futuro, a interface do sistema será otimizada para facilitar o cadastro e a gestão das informações de contato diretamente pela interface gráfica.

Conclusões

O modelo de previsão de pragas se mostrou eficaz na identificação delas a partir de imagens da Internet, assim como no compartilhamento de informações para o usuário. O Roboflow se mostrou uma ferramenta muito útil na criação de modelos de visão computacional, tendo uma interface amigável ao usuário e sua fácil integração com diversas linguagens de programação.

Portanto, para os próximos passos, pretende-se realizar a atualização do sistema para poder ser usado localmente, sem precisar acessar a Internet, e a adição de novas imagens reais de alta qualidade para aperfeiçoar o modelo e facilitar a identificação para casos e ângulos específicos. Além também de criar interfaces gráficas para aprimorar o acesso do usuário ao sistema, já que, na versão atual, qualquer acesso do agricultor, como cadastrar o e-mail e escolher qual versão do modelo utilizar, é feito a partir do Terminal.

Devido ao fato de não haver quaisquer lavouras de trigo na faculdade ou na região, não foi possível realizar testes concretos do algoritmo desenvolvido, o que complexifica o seu uso em escalas comerciais. É necessário conduzir experimentos em lavouras reais para validar a eficácia do algoritmo e verificar seu desempenho em condições de campo.

Apesar do projeto ainda não ter as características que o tornariam amigável ao usuário e um sistema de uso comercial em grande escala, ele cumpre sua função de identificação e comunicação com o agricultor.

Referências Bibliográficas

- Barbedo, J. G. A. (2022) Deep learning applied to plant pathology: the problem of data representativeness. *Tropical Plant Pathology*, v. 47, p. 85-94.
- Barbedo, J. G. A. (2018) Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Computers and Electronics in Agriculture*, v. 153, p. 46-53.
- Cesaro Júnior, T.; Rieder, F.; Di Domênico, J. R., Lau, D. (2022) InsectCV: A system for insect detection in the lab from trap images, *Ecological Informatics*, v. 67, 101516. <https://doi.org/10.1016/j.ecoinf.2021.101516>
- Costa, V. O. (2023) Pulgão no Trigo: ocorrência, danos, monitoramento e controle. *News O. Salim*. Disponível em: <https://news.osalim.com.br/agronegocio/pulgao-no-trigo-ocorrencia-danos-monitoramento-e-controle?uid=124420>.
- Embrapa (2024) *Busca de imagens*. Disponível em: <https://www.embrapa.br/busca-de-imagens>. Acesso em: 7 dez. 2024.
- Exame Solutions (2024) Agricultura de precisão: o que é e por que tem sido tão importante para a produção de alimentos. *Exame Agro*. Disponível em: <https://exame.com/agro/agricultura-de-precisao-o-que-e-e-por-que-tem-sido-tao-importante-para-a-producao-de-alimentos/>. Acesso em: 30 set. 2024.
- Ferentinos, K. P. (2018) Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture*, V. 145, P. 311-318.
- Lins, E. A.; Rodriguez, J. P. M.; Scoloski, S. I.; Pivato, J.; Lima, M. B.; Fernandes, J. M. C.; Pereira, P. R. V. S.; Lau, D.; Rieder, R. (2020) A method for counting and classifying aphids using computer vision. *Computers and Electronics in Agriculture*, v. 169, p. 105200. <https://doi.org/10.1016/j.compag.2019.105200>.
- Mohanty, S. P.; Hughes, D. P.; Salathé, M. (2016) Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, v. 7, article 1419.
- Zanini, E.; Zaions, D. F.; Zanatta, A. L.; Hölbig, C. A.; Pavan, W. (2021) Modelos de aprendizagem de máquina para predição da presença de desoxinivalenol em grãos de trigo por meio de análises multiespectrais. Congresso Brasileiro de Agroinformática (SBIAGRO), Porto Alegre: Sociedade Brasileira de Computação, p. 207-214. <https://doi.org/10.5753/sbiagro.2021.18392>.