Reconhecimento e contagem de afídeos em tempo real usando ESP32-CAM

Theilor Alessi[1], **Rafael Rieder**[2]

[1]Instituto de Tecnologia – Universidade de Passo Fundo (UPF)
BR 285 Km 292,7 - Campus I, São José, Passo Fundo, RS, 99052-900

{183956, rieder}@upf.br

Abstract. *This work proposes developing a rapid system for recognizing and counting aphids, pests that significantly harm agricultural productivity. Using an ESP32-CAM module, the project aims to demonstrate the ability of a simple solution to capture images and identify and count aphids in real time for preliminary decision-making through the AphidCV computer vision method and an artificial intelligence model of the YOLO architecture. Despite the hardware limitations, which made it impossible to process images and infer the model directly on the ESP32-CAM, we adapted the system to delegate these tasks to a personal computer. At the same time, the camera realized images' capture and transmission. The project showed the potential and limitations of low-cost devices and how to scale them to use in insect monitoring applications.*

Resumo. *Este trabalho propõe o desenvolvimento de um sistema rápido de reconhecimento e contagem de afídeos, pragas que prejudicam significativamente a produtividade agrícola. Utilizando um módulo ESP32-CAM, o projeto visa demonstrar a capacidade de uma solução simples capturar imagens e identificar e contar pulgões em tempo real, para uma tomada de decisão preliminar, por meio do método de visão computacional AphidCV e de um modelo de inteligência artificial da arquitetura YOLO. Apesar das limitações do hardware, que inviabilizaram o processamento de imagens e a inferência ao modelo diretamente na ESP32-CAM, o sistema foi adaptado para delegar essas tarefas a um computador pessoal, enquanto a câmera atua na captura e na transmissão de imagens. O projeto mostrou o potencial e as limitações que dispositivos de baixo custo tem, e como eles podem ser dimensionados, para serem utilizados em aplicações de monitoramento de insetos.*

1. Introdução

Os afídeos são uma das pragas mais comuns nas plantações. Eles podem danificar uma variedade de plantas, incluindo muitas árvores frutíferas e culturas importantes, como soja, arroz e outros grãos[Hussain et al. 2024]. Eles se reproduzem rapidamente e causam sérios danos às plantas, como deformação de folhas, caules e frutos, infecção (vetores virais), além de sérios danos à planta, causa sérios danos aos agricultores [Ng & Perry 2004].

O controle de afídeos pode ser realizado de várias maneiras, incluindo o uso de pesticidas químicos, predadores naturais, como joaninhas e vespas parasitoides, e práticas

de manejo integrado de pragas - que combinam diferentes métodos para controlar as populações de forma eficaz e sustentável [Parra 2002].

Nesse contexto, o monitoramento das populações e a identificação de afídeos por meio do manejo integrado podem fornecer dados importantes relacionados à dinâmica populacional e ao manejo integrado dessas pragas [Liu et al. 2016]. Para isso, são utilizadas armadilhas que capturam estes insetos, tornando possível rastrear e estudar sua flutuação populacional [De Cesaro Júnior et al. 2022]. E, para agilizar esse processo, ferramentas de software para gestão, contagem, mensuração e classificação automática de afídeos também são empregadas.

AphidCV é um software de contagem, mensuração e classificação automática de afídeos que emprega técnicas de processamento de imagens, visão computacional e *deep learning* [Lins et al. 2020, Rodriguez & Rieder 2020], e está integrado a uma plataforma web de monitoramento de insetos [Kirinus et al. 2023a]. Ele segue um protocolo rígido de coleta e digitalização de amostras de bioensaios com afídeos utilizando *scanners*, e também uma versão que utiliza a câmera de dispositivos móveis [Kirinus et al. 2023b]. Esses recursos agilizam o processo de monitoramento dos pesquisadores e a emissão de boletins de alerta aos agricultores.

Neste contexto, este trabalho apresenta um projeto de reconhecimento e contagem de afídeos em tempo real usando o módulo ESP32-CAM, uma placa de desenvolvimento ESP32 com uma câmera OV2640. A ideia é demonstrar a capacidade de uma solução simples capturar imagens e identificar e contar pulgões em tempo real, utilizando o método e um modelo de inteligência artificial do software AphidCV. Devido ao seu baixo custo, tamanho reduzido e baixo consumo de energia, a ESP32-CAM pode oferecer vantagens para uso em áreas onde o monitoramento e a detecção rápida de pulgões *in loco* seja independente de conexão de rede, para uma tomada de decisão preliminar.

Para tanto, o trabalho está organizado como segue: a Seção 2 mostra trabalhos relacionados a respeito da temática do estudo; a Seção 3 apresenta os materiais e métodos utilizados para concepção da abordagem proposta, bem como detalha o desenho do estudo de caso proposto; a Seção 4 divulga os resultados obtidos com uma avaliação preliminar, e discute as vantagens e limitações do estudo; e a Seção 5 apresenta as conclusões e os trabalhos futuros.

2. Trabalhos Relacionados

A pesquisa de [Saradopoulos et al. 2022] investigou o uso de dispositivos de rastreamento de pragas em áreas urbanas, aproveitando técnicas de computação avançadas para coletar e analisar dados de maneira eficiente e econômica. O objetivo da pesquisa era desenvolver um concurso geral de insetos que não se restringisse a uma espécie específica e abrangesse uma ampla gama de tamanhos corporais e padrões de asas. Para conseguir isso, ferramentas baratas como ESP32-CAM e técnicas de aprendizagem profunda são usadas para armazenar imagens de insetos em armadilhas pegajosas. O método proposto reduz a sobreposição de insetos nas imagens e facilita a criação de diversos dados.

Os resultados mostram a eficácia deste método, que permite a verificação de dados sem a necessidade de rotulagem extensa, criando imagens contendo números variados de insetos. Os autores concluíram que a solução é um sistema de monitoramento fácil de usar

e acessível que pode ser amplamente adotado pelos cidadãos e partes interessadas locais para mapear a fauna de insetos em áreas urbanas. Além disso, a utilização de tecnologias de comunicação de baixo consumo de energia, como o LoRa, ajuda a melhorar a eficiência energética dos equipamentos, tornando a monitorização remota e a automação em grande escala cruciais para a gestão sustentável das cidades.

O trabalho de [Reddyboina et al. 2024] explora o uso de tecnologias de aprendizado de máquina e Internet das Coisas (IoT) para identificar e gerenciar estrategicamente pragas para aumentar a eficiência das colheitas e reduzir o uso de pesticidas. O principal objetivo é desenvolver um sistema que utilize inteligência artificial para detectar e classificar pragas em culturas como milho, soja e trigo. Imagens de insetos usando marcadores de caixa delimitadora para indicar sua localização e a arquitetura EfficientDet-Lite2 são usadas para construir o modelo de detecção de objetos. Isso permite que o produto seja executado em hardware limitado, como Raspberry Pi. O treinamento do modelo inclui técnicas de aprimoramento de dados, como rotação e escalonamento, para aumentar a variabilidade e robustez dos dados.

Os resultados mostram que o sistema atinge alta precisão na detecção e classificação de insetos com tempo computacional razoável. A combinação de aprendizado de máquina e IoT permite que os agricultores enviem instantaneamente notificações sobre danos causados por insetos e folhas, bem como conselhos sobre o uso correto de pesticidas. Os autores afirmam que esta abordagem não só melhora a capacidade dos agricultores de lidar com infecções, mas também promove a sustentabilidade agrícola, reduzindo a dependência de pesticidas e promovendo uma agricultura responsável.

O projeto [Costa et al. 2023] fornece uma nova solução para monitoramento de pragas agrícolas usando IoT e visão computacional (VC). O objetivo era desenvolver um sistema capaz de detectar automaticamente moscas brancas em estufas de tomate, tornando o monitoramento de pragas independente, econômico e preciso. Câmeras de baixo custo são usadas para capturar imagens de armadilhas pegajosas, e modelos de aprendizagem profunda, como o YOLOv5, são usados para detectar insetos. O sistema é incorporado à rede IoT e as imagens capturadas são enviadas para a nuvem, operadas pelo modelo de venture capital, e disponibilizadas aos agricultores por meio de um aplicativo móvel.

Os resultados mostram que o modelo YOLOv5 (XLarge) alcançou uma precisão média ponderada (mAP) de 89,7% na detecção de moscas brancas. Além de contar as populações de insetos, a solução monitoriza continuamente o progresso das infestações de pragas, fornecendo aos agricultores dados instantâneos para tomarem decisões informadas. O aplicativo móvel facilita o acesso às informações, permitindo aos usuários monitorar de forma visual o progresso das pragas em sua área. O autor afirmou que a combinação de IoT e capital de risco, aliada a práticas agrícolas sustentáveis e eficientes, melhorou a qualidade da produção agrícola.

Por fim, o estudo [Cardoso et al. 2022] também demonstra o desenvolvimento de um sistema inteligente de rastreamento de insetos combinando IoT e VC – neste caso, a detecção e contagem geral de insetos pode ser aumentada. O sistema usa câmeras baratas anexadas a armadilhas adesivas para capturar imagens de insetos e enviá-las para a nuvem para processamento. Modelos de aprendizagem profunda, como R-CNN e YOLO, têm

sido usados para identificar e contar moscas brancas em imagens coletadas. A abordagem envolve a criação de um repositório robusto capturado em condições realistas para treinar o modelo VC e validar seu desempenho em situações do mundo real.

Além disso, os resultados mostram que a combinação de IoT e visão computacional melhora a precisão do rastreamento de insetos, tornando o sistema mais autônomo e menos dependente da intervenção humana. Fornece aos agricultores análises contínuas das populações de pragas ao longo do tempo por meio de um aplicativo móvel e acompanhe o desenvolvimento de pragas em cada campo por meio de uma interface personalizável. Além de reduzir os custos operacionais associados à monitorização manual, a solução poderia ajudar a reduzir o abuso de substâncias, dizem os autores. Outro destaque da pesquisa é a capacidade de fornecer informações instantaneamente e facilitar a tomada de decisões informadas, o que contribui para uma gestão agrícola eficaz.

Nosso projeto é baseado na utilização de uma placa ESP32-CAM para capturar e visualizar pulgões em tempo real através de um servidor, proporcionando um diferencial significativo no monitoramento de danos. Primeiro, o sistema é de fácil acesso e baixo custo, facilitando sua implementação em larga escala. Os recursos de análise instantânea identificam com rapidez e precisão as espécies monitoradas, fornecendo dados instantâneos para facilitar a tomada de decisões no manejo de pragas. A integração do ESP32-CAM com algoritmos de aprendizagem profunda pode melhorar a detecção e classificação de insetos e reduzir a necessidade de ferramentas de processamento de imagem de última geração. O objetivo é tornar o projeto proposto eficiente, escalável e adaptável, ajudando a estabelecer métodos de monitorização sustentáveis, alinhados com os princípios das cidades inteligentes e da agricultura moderna.

3. Materiais e Métodos

Neste projeto foi utilizado o microprocessador ESP32-CAM, equipado com firmware MicroPython, para capturar imagens em tempo real. A captura de imagens foi processada em um notebook Lenovo Ideapad 3, modelo 15ALC6, processador AMD Ryzen 7 5700U 4.3 GHz, memória RAM de 16GB, e unidade de processamento gráfico integrada AMD Radeon Graphics. Para a detecção e inferência de diferentes tipos de afídeos, utilizou-se um modelo de deep learning baseado na arquitetura YOLO (*You Only Look Once*), denominado AphidCV, para reconhecimento e classificação de afídeos.

A ESP32-CAM é um microcontolador da família ESP, altamente utilizada em sistemas IoT para captura de imagens e vídeo (Figura 1). Ela é composta pelo chip ESP32 e uma câmera de dois megapixels (OV2640), e se comunica com a IDE através de uma placa auxiliar, ESP32-CAM-MB. O dispositivo suporta cartão micro SD para gravação de dados, além de contar com flash em LED integrado e vários pinos GPIOs para conectar periféricos.

YOLO é um modelo estado-da-arte para detecção e segmentação de objetos em imagens, que utiliza uma rede neural convolucional como extrator de características (*feature extraction*) [Redmon et al. 2016]. Tem como suas principais características a alta precisão e excelente desempenho, permitindo a identificação de objetos de interesse em tempo real, mesmo em dispositivos de reduzido poder computacional.

AphidCV é um método que utiliza recursos de visão computacional e *deep learning* para automatizar a contagem, a classificação e a mensuração de afídeos, de dife-

Figura 1. ESP32-CAM. Fonte: [Eletrogate 2024].

rentes espécies de afídeos, em amostras de bioensaios [Lins et al. 2020]. Cada elemento de amostra é categorizado em quatro classes, sendo três estádios de desenvolvimento do afídeo (ninfas, ápteros e alados) e uma categoria relacionada a exosqueletos e detritos (falsos), como mostra a Figura 2.

Especificamente para este estudo, considerou-se apenas o uso de modelo AphidCV para identificação da espécie *Rhopalosiphum padi* (Rp), gerado com a arquitetura YOLOv8 [Jocher et al. 2023], versão "nano" (YOLOv8n).

Figura 2. Exemplos de afídeos da espécie *Rp* para ilustrar as classes em cada conjunto, na ordem (esquerda para direita): ninfa, áptero, alado e falso.

MicroPython[1] é uma implementação da linguagem de programação Python 3 otimizada para rodar em microcontroladores e sistemas embarcados, que possui dentro de si uma seleção reduzida de bibliotecas padrão do próprio Python. Por seu foco ser dispositivos com baixo consumo de energia e recursos limitados, como ESP32 e ESP8266, ele é uma solução compacta que utiliza em sua execução apenas 256KB de espaço para código e 16KB de memória RAM. Além disso, possui uma interface simples e intuitiva para programar sensores, atuadores e outros componentes.

Por ser capaz de rodar diretamente nos microcontroladores, o MicroPython elimina a necessidade de sistemas operacionais, sendo ideal para uso em aplicações de tempo real. Outro ponto forte é o seu leque de funcionalidades, tendo suporte nativo a comunicação via I2C, SPI, UART, Wi-Fi e Bluetooth, além de diversas bibliotecas para controle de hardware.

[1]Disponível em `https://micropython.org`.

A instalação do firmware MicroPython na ESP32-CAM foi realizada por intermédio do Thonny[2], um ambiente de desenvolvimento integrado (IDE) gratuito e de código aberto para Python. Ele permite programar para placas como ESP32 e ESP8266, e é compatível com diferentes plataformas operacionais.

A partir desses materiais, o objetivo inicial do projeto consiste em utilizar os recursos do MicroPython para executar o modelo YOLO no próprio microprocessador ESP32, capturando a imagem e realizando a inferência na própria placa, tornando-a um dispositivo único. Com isso, a ideia é ter um equipamento simples, portátil e de baixo custo, que possa ser utilizado para detecção e classificação rápida de afídeos, inicialmente em amostras de laboratórios em placas de Petri. A Figura 3 apresenta um esquemático da ideia de funcionamento da solução.

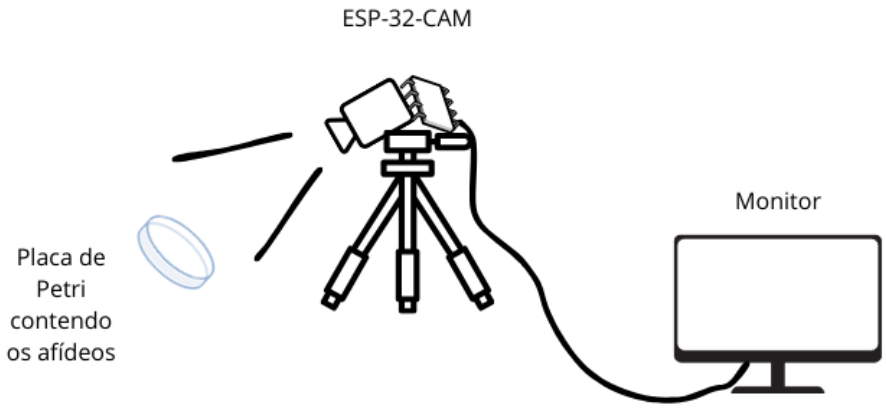


Figura 3. Esquemático da ideia inicial.

Para alcançar esse objetivo, foi instalado o sistema Micropython na ESP32-CAM, para programação direta do código de controle e acesso nativo aos módulos de câmera e Wi-Fi da placa, para processar as imagens diretamente no dispositivo. O modelo YOLO ficou hospedado em um cartão SD, o qual também armazena as capturas de imagens e respectivas inferências.

Para realizar os testes, a ESP32-CAM foi posicionada sobre um suporte ajustável, alinhado para uma área específica contendo a amostra com os afídeos a serem reconhecidos, exibindo posteriormente os resultados em um monitor conectado ao sistema (processo de captura da imagem e a identificação ocorrendo em tempo real). Em virtude de tempo de projeto, e por não ter acesso a um laboratório de entomologia, os testes consideraram imagens de alta resolução (1200 dpi) de placas de Petri com insetos, conforme padrão AphidCV [Lins et al. 2020].

A avaliação do sistema também considerou a comparação das imagens das placas analisadas pela ESP-32CAM com o software AphidCV executando em um computador pessoal, utilizando o mesmo modelo YOLO. A ideia é verificar a diferença de desempenho entre as abordagens, com base no número de afídeos detectados por classe.

[2]Disponível em `https://thonny.org`.

4. Resultados e Discussão

A partir da instalação inicial do MicroPython na ESP32-CAM, descobriu-se que o microprocessador não se encontrava acessível pela IDE, impossibilitando o envio de códigos para o mesmo. Para resolver este problema, foi necessário realizar uma modificação no arquivo "configuration.ini" da própria IDE com os seguintes parâmetros:

```
[ESP32]
dtr = False
rts = False
```

Essa modificação estabilizou a conexão entre as duas partes, permitindo o uso da IDE para rodar códigos diretamente na ESP32-CAM.

Após isso, tentou-se executar diferentes códigos-fonte para teste de técnicas de processamento de imagens aplicadas diretamente à imagens adquiridas pela câmera. Nesse ponto, descobriu-se que o MicroPython não oferece suporte para bibliotecas de visão computacional como OpenCV[3] e Pillow[4], utilizadas para manipulação de imagens. Desta forma, a ideia original de realizar pré-processamento de imagens no dispositivo ficou inviabilizada.

Outro ponto crítico para o projeto foi a tentativa de rodar o modelo YOLO selecionado para o projeto localmente na ESP32 - sem manipular a imagem. Após tentativas de instalação das dependências mínimas da YOLO, e de simplificação do modelo, a ESP32 não foi capaz de realizar as operações necessárias de inferência na imagem, em virtude da limitação de recursos de memória (apenas 4MB de RAM, desconsiderando o programa em execução). Apesar de existirem projetos compartilhados publicamente que executam modelos de inteligência artificial na ESP32, os mesmos fazem uso de modelos de complexidade extremamente reduzida e de baixa precisão, para não interferirem no desempenho da placa.

Como apresentado na Seção 3, o objetivo desse estudo era capturar, processar e classificar imagens diretamente no microprocessador ESP32-CAM, usando MicroPython e o modelo YOLO AphidCV. No entanto, isso se mostrou inviável devido à falta de bibliotecas compatíveis com as funções necessárias por parte do firmware (por possuir uma biblioteca reduzida), além da limitada capacidade de processamento e memória da placa. Essas limitações obrigaram a adaptar o projeto, como forma de reduzir a carga de trabalho no microprocessador.

Portanto, a reestruturação do projeto considerou a ESP32-CAM apenas como um dispositivo de captura de imagens, delegando a inferência ao modelo YOLO e a manipulação da imagem para um computador pessoal (PC). Essa mudança permitiu que os desafios impostos pelas restrições de hardware e software da ESP32-CAM fossem superados, aproveitando-se então da conectividade Wi-Fi disponibilizada nativamente pela placa para realizar a transmissão de dados em tempo real para o PC.

A ESP32-CAM com MicroPython foi então reprogramada para, desta vez, fazer uma transmissão de imagens via protocolo HTTP para o PC, o qual foi programado para requisitar frames dessa transmissão periodicamente. O software de captura considerava,

[3]Disponível em `https://opencv.org`.

[4]Disponível em `https://pillow.readthedocs.io/en/stable/`

na análise dos frames recebidos, a presença de uma placa de Petri, a partir da detecção de círculos na imagem - no caso, a placa é forma geométrica circular, no qual os afídeos estão distribuídos. Desta forma, procedia-se com a segmentação da imagem que, posteriormente, fazia o recorte de todos os objetos de interesse presentes na amostra, para inferência ao modelo YOLO AphidCV.

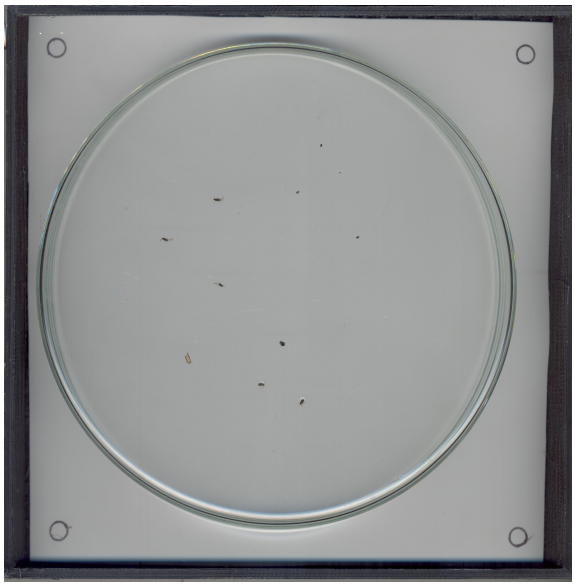
Os testes foram realizados de maneira muito similar ao projeto inicial. A ESP32 foi posicionada em um suporte, para garantir estabilidade, e direcionada a um monitor, aonde estava sendo apresentada uma imagem contendo os afídeos a serem analisados. Ao iniciar a transmissão, a ESP32 cria um servidor HTTP, do qual está configurado um PC localizado na mesma rede aguardando um streaming de vídeo. Ao receber os frames requisitados, inicia-se o processamento em tempo real no PC, exibindo como saída o streaming original capturado pela placa, e ao lado um streaming rotulado com as detecções realizadas pelo modelo. A Figura 4 ilustra a configuração do teste de bancada realizado.

Figura 4. Bancada de testes.

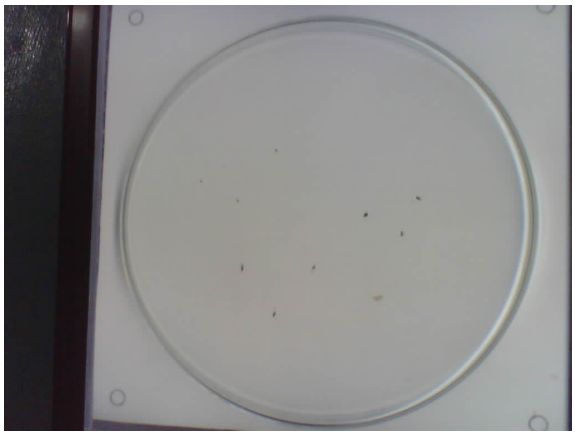
Devido a limitações de tempo do projeto, não foi possível testar a proposta em ambiente real, com amostra de afídeos dispostos em uma placa de Petri. Mesmo considerando imagens de alta resolução para simular as amostras de bioensaios, acabou ocorrendo certa perda de qualidade nos frames capturados. Esse fato ocorreu tanto pela captura estar sendo realizada de um monitor e não de uma bancada de testes verdadeira, quanto pela qualidade da imagem capturada pela própria câmera da ESP32, que não possui uma densidade de pixels elevada.

Por mais que a resolução da imagem possa ser considerada boa (a captura estava sendo realizada em HD 1024 X 768 pixels), ela não se equipara à imagem referência de laboratório no qual o método AphidCV está preparado para os processos de inferência

(imagem com resolução de 7703 x 7816 pixels) - como mostra a Figura 5. Essa situação acabou prejudicando o desempenho da detecção dos afídeos, por se tratarem de insetos de tamanho muito pequeno, e que um modelo inteligente exige uma qualidade de imagem com alto nível de detalhe para interpretar minúsculas diferenças de padrão presentes em cada estádio de desenvolvimento de um afídeo.



(a) Imagem de referência.

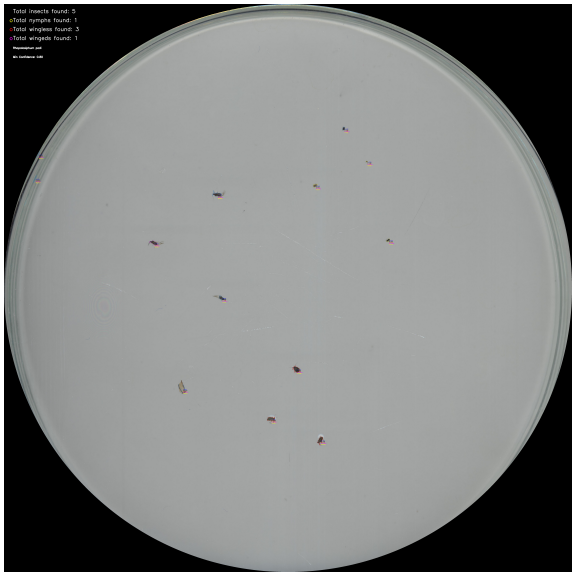


(b) Frame capturado pelo dispositivo.

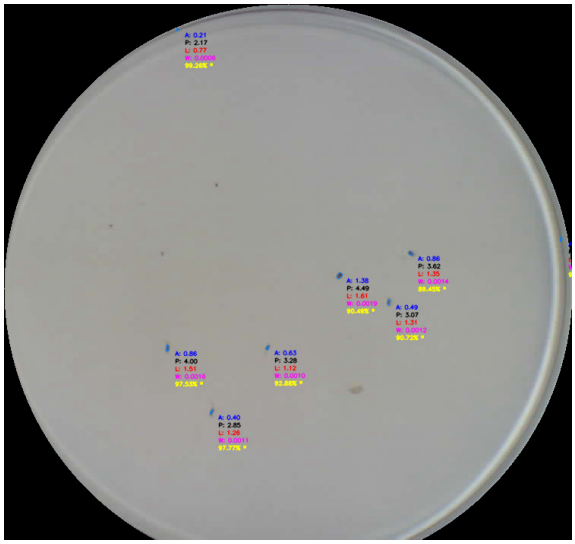
Figura 5. Comparação entre qualidade de imagem de referência e a de captura realizada pela ESP32-CAM, considerando a mesma amostra.

Mesmo sofrendo com a qualidade inferior ao ideal, percebe-se que o algoritmo consegue detectar a presença de objetos nas imagens recebidas por ele (rotulando-os como "falsos"), provando que é possível detectar que existem afídeos nos frames capturados (Figura 6). Porém, devido a falta de um detalhamento mais profundo na imagem, a classificação acaba não ocorrendo de maneira esperada. Acredita-se que, mesmo se a captura ocorresse diretamente em uma amostra de laboratório, num ambiente real, o resultado tende a ser similar, em virtude das características do objeto de interesse. Nesse sentido, pode ser interessante usar a abordagem proposta para recortar os objetos detec-

tados, e alimentar um retreinamento do modelo YOLO AphidCV, para que ele consiga aprender características dos insetos diminutos com qualidade inferior da imagem, que apresenta menos detalhes.



(a) Detecção esperada, a partir da imagem de referência.



(b) Detecção realizada a partir da imagem capturada pela ESP32.

Figura 6. Comparação de desempenho após o processo de inferência utilizando o modelo YOLO AphidCV.

Apesar dos desafios enfrentados, o projeto contribuiu de maneira significativa explorando o potencial e as limitações que dispositivos de baixo custo, como a ESP32-CAM, tem para serem utilizados em aplicações de monitoramento de insetos. A pesquisa mostrou que, por mais que haja restrições técnicas, soluções viáveis e acessíveis podem ser criadas combinando a captura e processamento de imagens em tempo real, abrindo caminho para que aplicações de alta tecnologia e baixo custo possam ser dimensionadas.

5. Conclusão

Este trabalho apresentou o desenvolvimento de uma solução para reconhecimento e contagem de afídeos em tempo real usando ESP32-CAM e o método AphidCV, considerando recursos de visão computacional e *deep learning*. Apesar dos desafios encontrados, o estudo mostrou que o projeto tem potencial para criar uma nova solução de execução em tempo real, de custo acessível, destinada ao monitoramento de pragas agrícolas. Para tanto, é preciso integrar dispositivos móveis de baixo custo e um computador que permita o processamento de alto desempenho.

Apesar dos desafios encontrados, como a incompatibilidade de bibliotecas e a limitação de recursos computacionais da ESP32-CAM para executar rotinas de visão computacional e modelos de detecção complexos, o projeto foi ajustado para combinar as capacidades da captura de imagem do dispositivo supracitado e o poder de processamento de um computador pessoal, mostrando que a detecção pode ser realizada *in loco*, tendo um suporte de rede para permitir a conexão remota entre equipamentos.

Os testes realizados mostraram que, por mais que a qualidade de imagem não seja a ideal, e a identificação ainda não seja precisa e como esperado, é possível detectar os objetos de interesse na imagem. Para melhorar a assertividade da detecção, é importante que um modelo inteligente seja gerado para dar suporte a imagens com resolução inferior. Nesse sentido, pode-se reavaliar a aplicabilidade do projeto e aprimorar o método proposto, especialmente considerando um ambiente real de captura como laboratórios de entomologia.

Com base nos resultados obtidos através deste projeto e nas limitações encontradas no decorrer do mesmo, as seguintes direções podem ser tomadas em possíveis trabalhos a serem feitos no futuro:

- Avaliar outras alternativas de microprocessadores, como Raspberry Pi AI Camera, para reduzir a dependência de um computador externo a placa realizando o processamento de imagens e inferência ao modelo, tentando manter o baixo custo;
- Avaliar outras câmeras suportadas pela ESP32, ou outras placas similares, com foco em aumentar a qualidade e densidade da imagem capturada, para capturar um nível maior de detalhe dos afídeos;
- Avaliar a transmissão de imagens capturadas pelo microprocessador para servidores na nuvem por redes IoT usando LoRa como protocolo de comunicação, como meio de disponibilizar em tempo real dados e análises da flutuação populacional de insetos aos pesquisadores, técnicos e produtores de culturas de cereais;
- Desenvolver protótipos autônomos, como drones, que utilizem esta tecnologia para capturar imagens em tempo real de diferentes áreas da lavoura, possibilitando o acesso a áreas mais remotas da lavoura e ampliando a área de cobertura do monitoramento;
- Adaptar e retreinar os modelos do AphidCV para suportar imagens em diferentes resoluções, como forma de considerar diferentes dispositivos de aquisição de imagens e melhorar os desempenhos de classificação de afídeos.

Referências

Cardoso, B., Silva, C., Costa, J., & Ribeiro, B. (2022). Internet of things meets computer vision to make an intelligent pest monitoring network. *Applied Sciences*, 12(18):9397.

DOI: https://doi.org/10.3390/app12189397.

Costa, D., Cardoso, B., Silva, C., Costa, J., & Ribeiro, B. (2023). Intelligent system for automatic whitefly detection in tomato greenhouses. In *2023 6th Experiment@ International Conference (exp. at'23)*, pages 29–30. IEEE. DOI: https://doi.org/10.1109/EXP.AT2358782.2023.10546195.

De Cesaro Júnior, T., Rieder, R., Di Domênico, J. R., & Lau, D. (2022). Insectcv: A system for insect detection in the lab from trap images. *Ecological Informatics*, 67:101516. DOI: https://doi.org/10.1016/j.ecoinf.2021.101516.

Eletrogate (2024). Introdução ao ESP32-CAM. Disponível em https://blog.eletrogate.com/introducao-ao-esp32-cam/. Acesso em: 11 nov. 2024.

Hussain, D., Saleem, U., Asrar, M., Saleem, M., Ghaffar, A., Hussain, S., Shaheen, T., & Saleem, M. (2024). A review on identification characters and ipm of wheat aphid by using non-conventional methods. *International Journal of Tropical Insect Science*, 44(2):419–432. DOI: https://doi.org/10.1007/s42690-024-01162-3.

Jocher, G., Qiu, J., & Chaurasia, A. (2023). Ultralytics YOLO. Disponível em https://github.com/ultralytics/ultralytics. Acesso em: 11 nov. 2024.

Kirinus, N. W., Pessolano, E. G., De Cesaro Júnior, T., Lau, D., & Rieder, R. (2023a). Aphidcv 3.0: Abordagem web integrada a uma plataforma de monitoramento de insetos. In *Anais do III Congresso de Tecnologia da Informação do IFSul - Passo Fundo*, pages 41–49. IFSul. Disponível em: https://periodicos.ifsul.edu.br/index.php/ctipf/article/view/3382.

Kirinus, N. W., Taca, B. S., Iora, M. S., Rieder, R., De Cesaro Júnior, T., & Lau, D. (2023b). Aphidcv mobile: ferramenta para classificação e contagem de afídeos em dispositivos móveis. In *Anais Estendidos do XXXVI Conference on Graphics, Patterns and Images*, pages 144–147. SBC. DOI: https://doi.org/10.5753/sibgrapi.est.2023.27468.

Lins, E. A., Rodriguez, J. P. M., Scoloski, S. I., Pivato, J., Lima, M. B., Fernandes, J. M. C., da Silva Pereira, P. R. V., Lau, D., & Rieder, R. (2020). A method for counting and classifying aphids using computer vision. *Computers and Electronics in Agriculture*, 169:105200. DOI: https://doi.org/10.1016/j.compag.2019.105200.

Liu, T., Chen, W., Wu, W., Sun, C., Guo, W., & Zhu, X. (2016). Detection of aphids in wheat fields using a computer vision technique. *Biosystems Engineering*, 141:82–93. DOI: https://doi.org/10.1016/j.biosystemseng.2015.11.005.

Ng, J. C. & Perry, K. L. (2004). Transmission of plant viruses by aphid vectors. *Molecular plant pathology*, 5(5):505–511. DOI: https://doi.org/10.1111/j.1364-3703.2004.00240.x.

Parra, J. R. P. (2002). *Controle biológico no Brasil: parasitóides e predadores*. Editora Manole Ltda. ISBN: 978-8520415542.

Reddyboina, A., Kumar, T. P., Varshitha, S., & Reddy, M. N. (2024). Smart agriculture by leveraging ai and iot for intelligent insect identification and control. In

2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), pages 1–5. IEEE. DOI: https://doi.org/10.1109/ICITEICS61368.2024.10625323.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788. DOI: https://doi.org/10.1109/CVPR.2016.91.

Rodriguez, J. P. M. & Rieder, R. (2020). Aphidcv 2.0: uma nova abordagem de classificação, contagem e mensuração de afídeos. In *Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 159–162. SBC. DOI: https://doi.org/10.5753/sibgrapi.est.2020.13001.

Saradopoulos, I., Potamitis, I., Ntalampiras, S., Konstantaras, A. I., & Antonidakis, E. N. (2022). Edge computing for vision-based, urban-insects traps in the context of smart cities. *Sensors*, 22(5):2006. DOI: https://doi.org/10.3390/s22052006.