

Inteligência artificial para monitoramento automatizado de insetos em armadilhas agrícolas

Diogo Yusuke Iwamoto¹

Marco Antônio Gusmão Carvalho²

¹ Discente em Big Data no Agronegócio, FATEC Shunji Nishimura, Pompeia-SP.

² Docente do curso Big Data no Agronegócio, FATEC Shunji Nishimura, Pompeia-SP.

RESUMO

Resumo: Pragas de insetos, como os tripses (Ordem Thysanoptera), representam uma séria ameaça à agricultura, causando danos diretos e transmitindo vírus em culturas como soja, batata, tomate e cebola. Sua detecção precoce é desafiadora devido ao seu tamanho reduzido e aos sintomas tardios, podendo levar a perdas de até 80% na produtividade. Atualmente, o monitoramento é feito manualmente, com contagem visual por lupas ou microscópios, um processo lento e propenso a erros. Diante disso, este projeto, desenvolvido pelo grupo Agrosense no Laboratório de Monitoramento e Proteção de Plantas (LMPP) da Fatec de Pompeia, teve como objetivo principal desenvolver um sistema automatizado baseado em inteligência artificial para identificar e monitorar tripses em armadilhas adesivas, agilizando o manejo integrado de pragas. A metodologia consistiu no treinamento de um modelo de reconhecimento de imagens utilizando dados de campo, com insetos capturados em armadilhas, processados por algoritmos de visão computacional. Os resultados demonstraram uma precisão de 86% na detecção, sendo a última média do treinamento, mostrando que pode ser mais eficiente do que os métodos tradicionais. Conclui-se que a automação do monitoramento de pragas, como proposta neste trabalho, oferece uma alternativa mais rápida e confiável, com potencial para reduzir perdas agrícolas e otimizar o controle de infestações. Futuros desenvolvimentos podem expandir a aplicação do sistema para outras pragas, ampliando seu impacto na agricultura de precisão.

Palavras-chave: tripes; monitoramento automatizado; inteligência artificial; agricultura de precisão; controle de pragas.

1 INTRODUÇÃO

Os insetos-praga representam um dos maiores desafios para a agricultura moderna, causando perdas anuais estimadas em até 40% da produção global de alimentos (OERKE, 2006). Dentre essas pragas, os tripes (Ordem Thysanoptera) destacam-se por sua capacidade de danificar culturas economicamente importantes, como soja, tomate, batata, pimentão e cebola, tanto através da alimentação direta quanto pela transmissão de vírus fitopatogênicos (RILEY *et al.*; 2011).

Espécies como *Frankliniella occidentalis* e *Thrips tabaci* são particularmente prejudiciais, podendo reduzir a produtividade em até 80% em casos de infestações severas (REITZ, 2009).

Tripos são insetos de corpo cilíndrico, pequeno e estreito, apresentam dois pares de asas franjadas e aparelho bucal sugador raspador, graças à adaptação de duas maxilas e uma mandíbula. São insetos polívoros e as fêmeas adultas são de maior tamanho (1,2 mm) que os machos (0,9 mm) (SALAS, 2017).

É considerada uma praga silenciosa que somente é notada quando a sua população se torna bastante elevada, tendo causado grandes problemas à cultura, já apresentando sintomas como área esbranquiçada ou prateada, trazendo em consequência a dificuldade de crescimento das plantas (MOURA, 2013).

O monitoramento convencional desses insetos, baseado em métodos manuais de contagem em armadilhas adesivas, apresenta limitações significativas: é laborioso, demanda expertise taxonômica e está sujeito a erros de amostragem (BINNS & NYROP 1992). Essas limitações retardam a tomada de decisão no manejo integrado de pragas (MIP), agravando os danos às culturas (PEDIGO; RICE, 2009). A falta de ferramentas precisas e ágeis para detecção precoce tem sido um obstáculo crítico para o controle eficiente dessas pragas (PIMENTEL, 2009).

Nos últimos anos, a agricultura de precisão tem incorporado soluções baseadas em inteligência artificial (IA) para otimizar o monitoramento de pragas (KAMILARIS & PRENAFETA-BOLDÚ, 2018). Técnicas de visão computacional e aprendizado de

máquina vêm sendo aplicadas com sucesso na identificação automatizada de insetos, como demonstrado por estudos com mosca-branca (DING; TAYLOR, 2016) e percevejos (BARBEDO, 2018).

Entretanto, para tripses - insetos menores (1-2 mm) e morfologicamente complexos - ainda há carência de sistemas robustos de detecção automatizada, especialmente em condições reais de campo (ESPINOZA ET AL., 2020).

O projeto desenvolvido é uma inteligência artificial, treinada por imagens do inseto tripses, que foram capturados em armadilhas adesivas. Com o objetivo de automatizar a detecção do inseto alvo, que possui um tamanho milimétrico, não sendo possível sua identificação a olho nu, assim, resultados mais rápidos e precisos serão obtidos, com uma redução na taxa de erros para um controle de pragas mais eficiente e ágil.

2 MATERIAIS E MÉTODOS

2.1. Área de Estudo e Coleta de Dados

A iniciativa foi desenvolvida pelos alunos da Fatec de Pompeia-SP, localizada na Fundação Shunji Nishimura de Tecnologia, através do laboratório de pesquisa LMPP (Laboratório de Monitoramento e Proteção de Plantas) que formou o grupo Agrosense.

O objetivo é criar uma ferramenta capaz de monitorar a lavoura, reconhecendo os tipos de insetos presentes, avaliando o nível de dano e assim ajudando os produtores e pesquisadores com mais agilidade e eficiência.

De início, o grupo teve a orientação do pesquisador Dr. Fernando J. Sanhueza Salas do Instituto Biológico de São Paulo, que direcionou e introduziu os primeiros conhecimentos sobre pragas na cultura da batata. O inseto escolhido para ser trabalhado foi o tripses, considerando ser uma praga desafiadora para investigação e prejudicial, não só para a batata, mas também para diversas outras culturas.



Figura 1 - Imagem do inseto tripses.

Fonte: <https://safraviva.com.br/tripes/>



Figura 2 - Armadilha adesiva posicionada para coleta de inseto

Fonte: Grupo GEAD Agrosense.

Os insetos-praga foram coletados na horta da Fundação, através do uso de armadilhas adesivas da cor amarela, sendo instaladas na posição vertical, fixas de forma aérea, como mostra a figura 2. Foram instaladas 10 armadilhas da marca BioTrap, expostas por 7 dias, realizando-se a troca das armadilhas após esse período. Assim, foram utilizadas 30 armadilhas em um período de 21 dias.

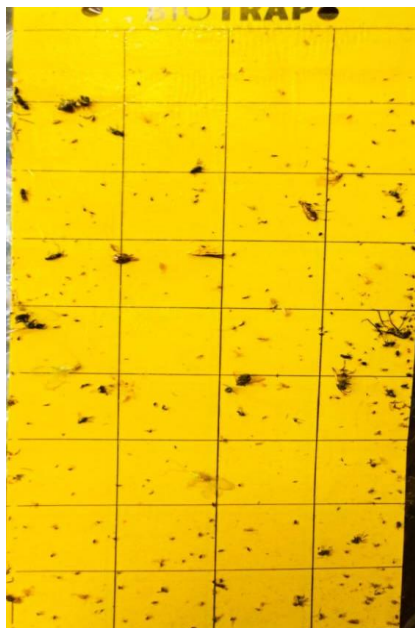


Figura 3 - Armadilha com insetos após 7 dias.
Fonte: Grupo GEAD Agrosense.

2.2 Pré Processamento de imagens

O grupo conseguiu capturar 244 imagens do inseto tripe nas armadilhas adesivas, com o uso de um microscópio digital (figura 4), armazenando no Google Drive. Em seguida foram criadas subpastas de treino, teste e validação, organizando as amostras e os labels, separando-os de modo que ficassem 75% das imagens para treino, 10% para teste e 15% para validação, valores normalmente utilizados para treinar um algoritmo, fornecendo a maior parte para treinamento do algoritmo.



Figura 4 - Imagem do Microscópio digital utilizado para capturar as imagens.
Fonte: Google Colab grupo GEAD Agrosense.

A execução do modelo de aprendizado foi realizada no ambiente de programação na nuvem Google Colab com a linguagem Python, utilizando-se as imagens separadas e organizadas no Drive, colocando 184 imagens em treino, 24 para teste e 36 para validação, conforme mostrado na figura 5.

```
[ ] import os

# Caminho da pasta principal
base_path = '/content/drive/MyDrive/AgroSense-V4-B_0.22'

# Subpastas
folders = ['train', 'test', 'val']
subfolders = ['images', 'labels']

# Dicionário para armazenar as contagens
counts = {folder: {subfolder: 0 for subfolder in subfolders} for folder in folders}

# Função para contar arquivos
def count_files(folder_path):
    if os.path.exists(folder_path):
        return len([f for f in os.listdir(folder_path) if os.path.isfile(os.path.join(folder_path, f))])
    else:
        print(f"Aviso: O caminho '{folder_path}' não foi encontrado e será ignorado.")
        return 0

# Contar os arquivos em cada subpasta
for folder in folders:
    for subfolder in subfolders:
        path = os.path.join(base_path, folder, subfolder)
        print(f"Verificando {path}...") # Para depuração
        counts[folder][subfolder] = count_files(path)

# Exibir resultados
for folder in counts:
    print(f"\nPasta {folder}:")
    for subfolder in counts[folder]:
        print(f"    {subfolder}: {counts[folder][subfolder]} arquivos")

↳ Verificando /content/drive/MyDrive/AgroSense-V4-B_0.22/train/images...
Verificando /content/drive/MyDrive/AgroSense-V4-B_0.22/train/labels...
Verificando /content/drive/MyDrive/AgroSense-V4-B_0.22/test/images...
Verificando /content/drive/MyDrive/AgroSense-V4-B_0.22/test/labels...
Verificando /content/drive/MyDrive/AgroSense-V4-B_0.22/val/images...
Verificando /content/drive/MyDrive/AgroSense-V4-B_0.22/val/labels...

Pasta train:
  images: 184 arquivos
  labels: 184 arquivos

Pasta test:
  images: 24 arquivos
  labels: 24 arquivos

Pasta val:
  images: 36 arquivos
  labels: 36 arquivos
```

Figura 5 - Conectando e inserindo as imagens e labels nas pastas do Drive.

Fonte: Google Colab grupo GEAD Agrosense.

2.3 Modelagem com Deep Learning

Antes do algoritmo ser treinado, as imagens foram demarcadas por quadrados denominados caixas delimitadoras (Bounding Boxes), através do software de visão computacional Roboflow. Com as figuras carregadas na ferramenta, os alunos detectaram os tripes em cada imagem e fizeram a demarcação em formato geométrico (figura 8).

Com as imagens demarcadas, foi utilizado um modelo de Rede Neural Convolutacional (CNN) YOLO (You Only Look Once) em sua versão 8, para a detecção de objetos em tempo real, o reconhecimento de tripes através das imagens quadriculadas,

a classificação do inseto através da cor, textura e padrão visual, sendo configuradas e instaladas as dependências no Google Colab para execução.



Figura 6 - Demarcação dos tripes com a ferramenta Roboflow.
Fonte: GEAD Agrosense.

2.4 Métrica de avaliação

Para se visualizar o desempenho do aprendizado na detecção com o YOLO, através da ferramenta Wandb (Weight & Biases), as métricas foram carregadas a cada época de treinamento, colocando-se os resultados de loss (perdas do algoritmo), trazendo-se os parâmetros de forma gráfica e contínua durante a execução.

Foram exibidos em forma de gráficos, importando em Python a biblioteca Matplotlib, com o recurso Pyplot, que oferece uma interface gráfica dos dados fornecidos, junto com o Seaborn, recurso que exhibe o gráfico após as avaliações finalizadas.

As métricas de avaliação foram:

- 1- Box Loss (Perda de localização)
- 2- Classification Loss (Perda de classificação)
- 3- Detection Loss (Perda de detecção)

3 RESULTADOS E DISCUSSÕES

3.1 Transformação de imagens

As imagens foram convertidas no formato RGB, para serem exibidas com a biblioteca Matplotlib, que exibe as imagens graficamente, e logo carregou-se para visualização de 20 imagens para teste de execução, exibido na figura 7.

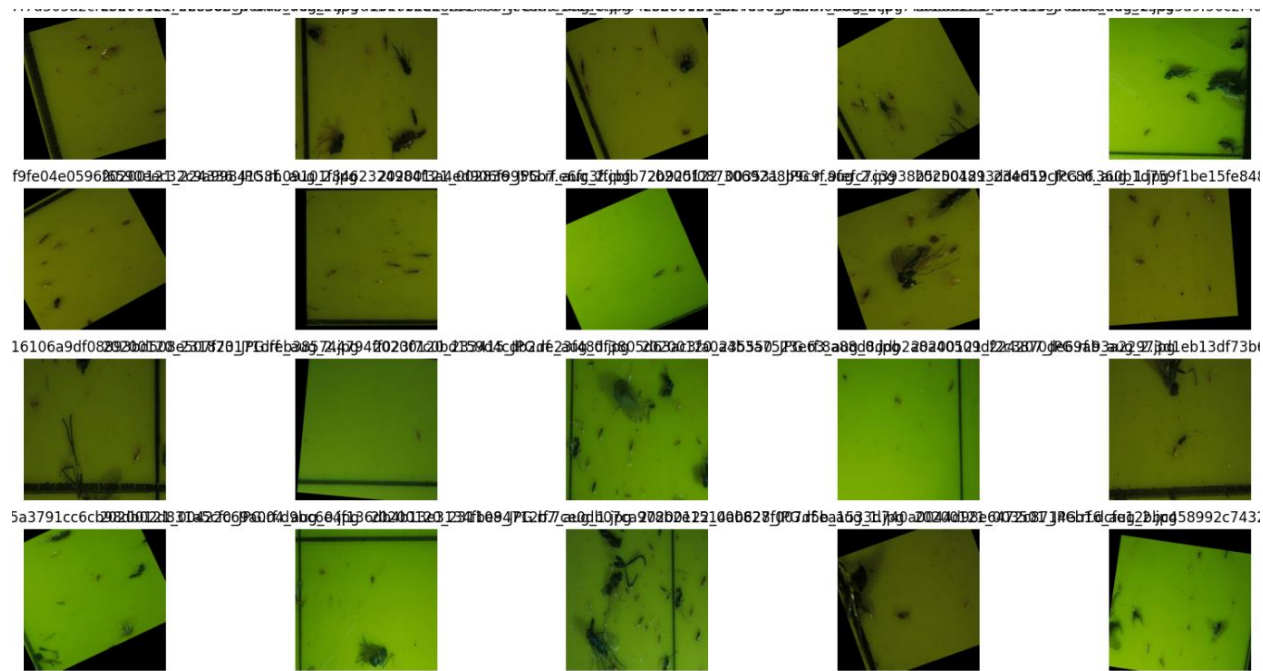


Figura 7 - Exibindo as imagens em RGB.
Fonte: Google Colab grupo GEAD Agrosense.

3.2 Treinamento do modelo de IA

Para o treinamento do modelo, na plataforma Colab do Google, foi iniciado o YOLO V8 e configurado o Wandb para rastrear e visualizar as métricas de treinamento de forma mais interativa. Foi criada uma pasta para salvar os resultados do treino e um diretório onde as imagens de validação foram copiadas.

```
# Inicializar o modelo YOLOv8
model = YOLO('yolov8m.yaml') # Usando um modelo YOLOv8 maior

# Caminho do Google Drive para salvar os resultados
drive_project_path = '/content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain'
os.makedirs(drive_project_path, exist_ok=True) # Criar a pasta no Google Drive, se não existir

# Criar pasta para armazenar as imagens de validação
drive_val_batch_path = os.path.join(drive_project_path, 'ResultsValBatch')
os.makedirs(drive_val_batch_path, exist_ok=True)

# Configuração do projeto WandB
project_name = 'yellow_sticky_modelV4-22-2025'
wandb_dir = '/content/ImagesResultsTrain' # Diretório local para WandB, se necessário

# Certifique-se de que o diretório local também exista
os.makedirs(wandb_dir, exist_ok=True)

# Autenticação no WandB (apenas para o primeiro uso, precisará inserir a chave)
wandb.login()
```

Figura 8 - Iniciando o treinamento e configurações do Wandb e do drive no Colab.

Fonte: Google Colab Grupo GEAD Agrosense.

O resultado do treinamento foi avaliado através de parâmetros definidos, como:

- 1- 150 épocas (total de ciclos de treinos)
- 2- Tamanho redimensionado das imagens de 640px
- 3- Batch: 16 imagens processadas por vez
- 4- Salvamento de pesos a cada 10 épocas
- 5- Workers: 2 subprocessos para carregar os dados

```

# Treinamento do modelo
results = model.train(
    data='/content/drive/MyDrive/AgroSense-V4-B_0.22/data.yaml', # Caminho para o arquivo data.yaml
    epochs=150, # Número de épocas para o treinamento
    imgsz=640, # Tamanho da imagem
    batch=16, # Tamanho do batch
    save=True, # Salvar os pesos e o modelo
    save_period=10, # Salvamento dos pesos a cada N épocas
    workers=2, # Número de workers
    project=drive_project_path, # Diretório do projeto apontando para o Google Drive
    name='ImagesResultsTrain', # Nome da pasta para salvar os resultados
    exist_ok=True, # Permitir sobreposição de resultados anteriores
    amp=True, # Habilitar mixed precision training
    device='cuda' # Usar GPU, se disponível
)

# Copiar as imagens de validação para a nova pasta
def copy_val_images(source_folder, destination_folder):
    if os.path.exists(source_folder):
        for file_name in os.listdir(source_folder):
            if file_name.startswith("val_batch") and file_name.endswith(".jpg"): # Filtrando arquivos de validação
                src_path = os.path.join(source_folder, file_name)
                dst_path = os.path.join(destination_folder, file_name)
                shutil.copy2(src_path, dst_path)
                print(f"Copiado: {file_name}")

val_images_source = os.path.join(drive_project_path, 'ImagesResultsTrain') # Pasta onde as imagens são salvas
copy_val_images(val_images_source, drive_val_batch_path)

# Exibir links para acessar os resultados
wandb_link = f"https://wandb.ai/AgroSenseTeste/{project_name}" # Substitua USERNAME pelo seu usuário WandB
print(f"Resultados salvos em: {drive_project_path} no Google Drive.")
print(f"Imagens de validação salvas em: {drive_val_batch_path}.")
print(f"Acesse o WandB para ver os resultados: {wandb_link}")

```

Figura 9 - Treinamento do modelo.

Fonte: Google Colab Grupo GEAD Agrosense.

Foram utilizadas as imagens originais, ou seja, que foram capturadas no microscópio para não sobrecarregar o programa, evitando-se falhas na compilação do programa.

Através do treinamento, mostrado na figura 9, fica visível nos resultados mostrados abaixo na figura 10, que as métricas estavam altas nas primeiras épocas, já havendo redução a cada época, assim ficando com valor baixo ao longo do treinamento, o que mostrou que o modelo tem um grande potencial de aprender a localizar e reconhecer objetos desejados.

Alguns fatores influenciaram no resultado, sendo outros objetos que podem estar presentes na imagem, como sujeira, folhas, asas, sobreposição de insetos, e a forma de demarcação das caixas preditas, o que pode reduzir alguns resultados na acurácia do modelo.

```

using 2 datacenter workers
Logging results to /content/drive/MyDrive/AgroSense-V4-8_0.22/ImagesResultsTrain/ImagesResultsTrain
Starting training for 150 epochs...

```

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size	640:	12/12	[00:07<00:00, 1.62it/s]	all	36	93	0	0
Class	Images	Instances	Box(P	R	mAP50	mAP50-95):	100%	2/2	[00:02<00:00, 1.23s/it]					
1/150	6.83G	5.57	6.434	4.178	20	640:	100%	12/12	[00:07<00:00, 1.62it/s]	all	36	93	0	0
2/150	6.78G	5.253	5.629	3.891	14	640:	100%	12/12	[00:06<00:00, 1.74it/s]	all	36	93	0	0
3/150	6.83G	4.082	4.208	3.466	26	640:	100%	12/12	[00:06<00:00, 1.96it/s]	all	36	93	0	0
4/150	6.83G	3.317	3.151	2.97	25	640:	100%	12/12	[00:06<00:00, 1.82it/s]	all	36	93	0	0
5/150	6.86G	3.801	2.774	2.73	29	640:	100%	12/12	[00:05<00:00, 2.06it/s]	all	36	93	0	0
6/150	6.8G	2.738	2.539	2.575	24	640:	100%	12/12	[00:06<00:00, 1.89it/s]	all	36	93	0	0

Figura 10 - Treinamento em execução, exibindo os resultados das primeiras épocas.

Fonte: Google Colab Grupo GEAD Agrosense.

Após finalizado o treinamento, o modelo exibiu os resultados, indicando que foram utilizadas 36 imagens para validação do aprendizado, apontando que, nas imagens, o modelo detectou 93 tripes.

A quantidade tem a importância de indicar quanto conteúdo verdadeiro o modelo precisava encontrar; um número muito baixo, ou seja, poucas instâncias, poderia indicar underfitting, ou seja, que o modelo teve poucos exemplos para o aprendizado.

Os resultados obtidos foram:

- 1- Precisão de 82,5%, ou seja, através das métricas de desempenho para a detecção de objetos, feitas no formato de Bounding Boxes (caixas delimitadoras).
- 2- Acertividade de 82,5% para apontar os tripes das imagens.
- 3- Recall de 86%, o que significa que o modelo encontrou 86% dos tripes reais das imagens.
- 4- A média de precisão (MAP50) foi de 87%, ou seja, mostra que o modelo está detectando com boa precisão as localizações dos tripes.
- 5- MAP50-95, uma métrica mais rigorosa, exigindo uma precisão na sobreposição da caixa prevista (Demarcação do modelo) com a caixa real (Bounding Box), o desempenho caiu para 49,2%, indicando que não há tanta precisão da caixa de

marcação dos objetos, mas ainda consegue localizar o tripses, mesmo com a marcação não tão precisa.

```

150 epochs completed in 0.375 hours.
Optimizer stripped from /content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain/ImagesResultsTrain/weights/last.pt, 52.0MB
Optimizer stripped from /content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain/ImagesResultsTrain/weights/best.pt, 52.0MB

Validating /content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain/ImagesResultsTrain/weights/best.pt...
Ultralytics 8.3.75 Python-3.11.11 torch-2.5.1+cu124 CUDA:0 (Tesla T4, 15095MiB)
YOLOv8m summary (fused): 218 layers, 25,840,339 parameters, 0 gradients, 78.7 GFLOPs

```

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95)
all	36	93	0.825	0.86	0.87	0.492

```

Speed: 0.3ms preprocess, 12.5ms inference, 0.0ms loss, 1.4ms postprocess per image
Results saved to /content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain/ImagesResultsTrain
Copiado: val_batch1_pred.jpg
Copiado: val_batch1_labels.jpg
Copiado: val_batch0_labels.jpg
Copiado: val_batch0_pred.jpg
Resultados salvos em: /content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain no Google Drive.
Imagens de validação salvas em: /content/drive/MyDrive/AgroSense-V4-B_0.22/ImagesResultsTrain/ResultsValBatch.
Acesse o WandB para ver os resultados: https://wandb.ai/AgroSenseTeste/yellow\_sticky\_modelV4-22-2025

```

Figura 11 - Resultado do treinamento após as 150 épocas concluídas.

Fonte: Google Colab Grupo GEAD Agrosense.

3.3 Resultados do desempenho no aprendizado do modelo

Com o modelo treinado, já é possível a detecção automática na exibição de imagens, mostrado na figura 12.

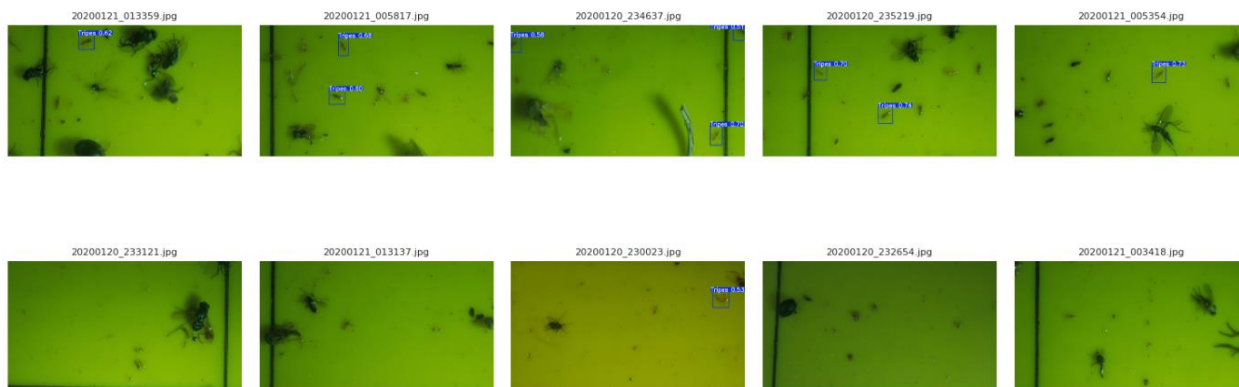


Figura 12 - Visualizando imagens com o treinamento.

Fonte: Google Colab Grupo GEAD Agrosense.

Para visualizar o desempenho do modelo durante o treinamento, foram criados gráficos com as bibliotecas Seaborn e Matplotlib, com o objetivo de indicar a taxa de

perdas de acordo com as épocas de treinamento, ou seja, a quantidade de vezes que o modelo foi treinando e aprendendo a localizar e detectar o inseto alvo.

Foram plotadas nos gráficos, métricas de Box Loss, classification Loss e Detection Loss, facilitando a visualização dos resultados durante as épocas de treinamento do modelo, conforme a figura 13.

```
# 1. Box Loss
if 'train/box_loss' in metrics_df.columns:
    sns.lineplot(data=metrics_df, x=metrics_df.index, y="train/box_loss", ax=axes[0, 0], color='b', linewidth=2)
    axes[0, 0].set_title("Box Loss (Perda de Classificação)")
    axes[0, 0].set_xlabel("Épocas")
    axes[0, 0].set_ylabel("Box Loss")

# 2. Classification Loss
if 'train/cls_loss' in metrics_df.columns:
    sns.lineplot(data=metrics_df, x=metrics_df.index, y="train/cls_loss", ax=axes[0, 1], color='g', linewidth=2)
    axes[0, 1].set_title("Classification Loss (Perda de Localização)")
    axes[0, 1].set_xlabel("Épocas")
    axes[0, 1].set_ylabel("Classification Loss")

# 3. Detection Loss
if 'train/dfl_loss' in metrics_df.columns:
    sns.lineplot(data=metrics_df, x=metrics_df.index, y="train/dfl_loss", ax=axes[1, 0], color='r', linewidth=2)
    axes[1, 0].set_title("Detection Loss (Perda de Detecção)")
    axes[1, 0].set_xlabel("Épocas")
    axes[1, 0].set_ylabel("Detection Loss")
```

Figura 13 - Inserindo as métricas de avaliação nos gráficos.

Fonte: Google Colab Grupo GEAD Agrosense.

De acordo com os resultados dos gráficos mostrados nas figuras abaixo 14 e 15, o treinamento obteve um ótimo desempenho, mostrando-se evolução a cada época.

Visualizando-se os gráficos, observa-se que a taxa de perda de todos os parâmetros começou alta e reduziu drasticamente já nas primeiras épocas, diminuindo a cada repetição do treinamento, mostrando a capacidade do modelo aprender a localizar e detectar os tripes demarcados.

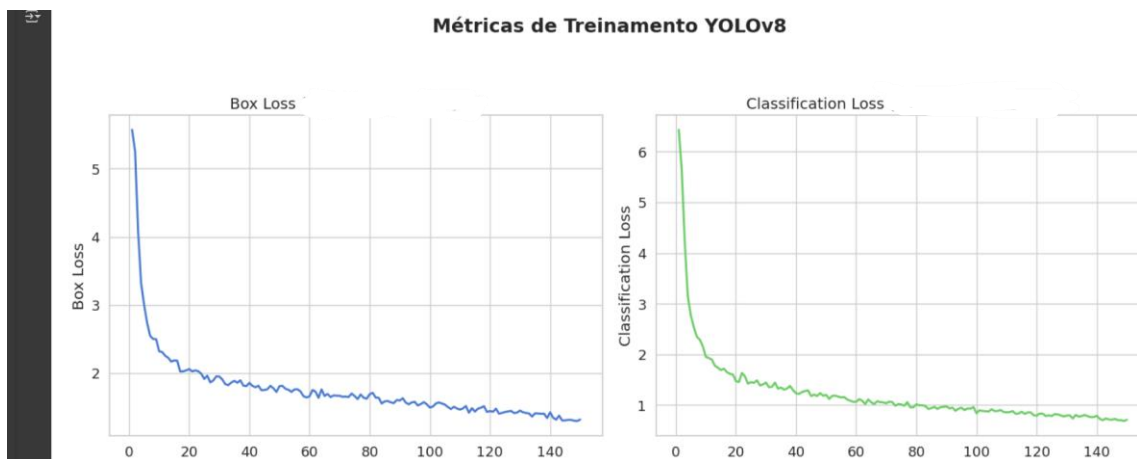


Figura 14 - Gráficos de Box Loss e Classification Loss.

Fonte: Google Colab Grupo GEAD Agrosense.

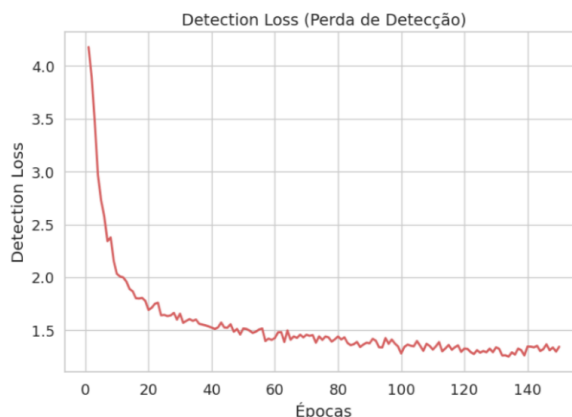


Figura 15 - Gráfico de Detection Loss.

Fonte: Google Colab Grupo GEAD Agrosense.

Explicando sobre as métricas de avaliação:

- 1- **Box Loss:** Mostrou que o modelo foi aprendendo a localizar as caixas delimitadoras nas imagens, já nas primeiras épocas de treinamento, reduzindo drasticamente a taxa de erros de localização das coordenadas das caixas.
- 2- **Classification Loss:** O modelo, através do treinamento, conseguiu aprender a classificar o tripes dentro da caixa delimitadora, reduzindo a taxa de erros em localizar as demarcações das caixas, e diminuindo os erros na classificação do objeto em questão.
- 3- **Detection Loss:** Aponta que o modelo aprendeu a localizar e reconhecer os tripes nas imagens, através da soma dos parâmetros de Box Loss e Classification Loss.

```

# Verifique se a métrica metrics/mAP50(B) está presente no DataFrame de métricas
if 'metrics/mAP50(B)' in metrics_df.columns:
    # Pegue a precisão final da última época
    final_map = metrics_df['metrics/mAP50(B)'].iloc[-1] * 100 # Converta para porcentagem
    margem_erro = 100 - final_map # Margem de erro é o complemento em relação a 100%

    print(f"Precisão Geral do Modelo: {final_map:.2f}%")
    print(f"Margem de Erro do Modelo: {margem_erro:.2f}%")
else:
    print("A métrica 'metrics/mAP50(B)' não foi encontrada no arquivo de métricas.")

```


 Precisão Geral do Modelo: 86.98%
 Margem de Erro do Modelo: 13.02%

Figura 16 - Precisão geral do modelo e margem de erros.

Fonte: Google Colab Grupo GEAD Agrosense.

Conforme a figura 16, o modelo obteve um resultado de 86,98% de precisão, valor esse retirado da última época de treinamento, subtraindo-se a diferença, gerou-se uma margem de erros de 13,02%, ou seja, o modelo está acertando em torno de 87 a cada 100 previsões, mas pode errar cerca de 13% das vezes. Tal resultado pode ser considerado aceitável no auxílio para identificação e contagem de insetos alvo, mas pode ser melhorado no ponto de vista comercial, com a utilização da IA em situações reais de campo, para previsão e alerta de infestações, que possibilitem rápida tomada de decisão e resolução do problema.

Em suma, a inteligência artificial desenvolvida mostra ser capaz de detectar os trips capturados nas armadilhas adesivas, com possibilidade de melhora na precisão da sobreposição das caixas previstas e real.

4 CONSIDERAÇÕES FINAIS

Através do modelo de treinamento, a inteligência artificial se mostrou eficaz para aprender a localizar e reconhecer os trips capturados nas armadilhas adesivas, mostrados em imagens feitas pelo microscópio digital.

Porém, o modelo ainda pode reduzir a margem de erros, chegando próxima de zero, e a precisão próxima de 100. As dificuldades a serem transpostas são: poucas amostras para o treinamento da IA, o programa no ambiente virtual não suporta compilar grandes quantidades de imagens, as figuras apresentavam sujeiras e folhas sobrepostas, e a falta de equipamento com boa resolução de imagem para melhor visualização do objeto.

Para trabalhos científicos e auxílio de profissionais na identificação e contagem de insetos, pode-se dizer que o modelo será útil para agilizar e reduzir erros em relação ao trabalho manual.

Os projetos desenvolvidos no LMPP são feitos por alunos que visam um aprendizado prático e profundo na pesquisa de monitoramento agrícola e desenvolvimento de novas tecnologias, e que continuam trabalhando no projeto Agrosense visando obter um modelo com maior precisão e consequente redução da

margem de erros, através do fornecimento de maior quantidade de treinamentos com imagens, e melhorias na marcação das caixas reais para a localização dos insetos.

Logo, este método poderá ser utilizado para identificação de outros tipos de insetos-praga, como mosca branca, pulgão e bicho mineiro, bem conhecidos por danificar culturas de grande importância no Brasil, e assim ajudar profissionais da agricultura e da pesquisa a obterem resultados mais rápidos e precisos. Num futuro breve, o modelo com precisões mais altas certamente será de grande valia na tomada de decisão para aplicações em campo.

REFERÊNCIAS

- BINNS, M. R.; NYROP, J. P. Sampling Population for the purpose of IPM decision making. *Annual Review of Entomology*, v.37, n.1, p 427-453, 1992.
- DING, W.; TAYLOR, G. Automatic moth detection from trap images for pest management. *Computers and Electronics in Agriculture*, v. 123, p. 17-28, 2016.
- ESPINOZA, K. et al. Deep learning for insect identification in crop protection. *Precision Agriculture*, v. 21, n. 2, p. 280-296, 2020.
- KAMILARIS, A.; PRENAFETA-BOLDÚ, F. X. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, v. 147, p. 70-90, 2018.
- MOURA, Alexandre Pinho. Trips: a praga silenciosa. *Revista Campo & Negócios HF*, Uberlândia, Ano 8, n. 93, p. 28-30, 2013.
- OERKE, E. C. Crop losses to pests. *The Journal of Agricultural Science*, v. 144, n. 1, p. 31-43, 2006.
- PEDIGO, L. P.; RICE, M. E. *Entomology and pest management*. 6. ed. Upper Saddle River: Pearson Prentice Hall, 2009.
- PIMENTEL, D. Pesticides and pest control. In: PESHIN, R.; DHWAN, A. K. (Ed.). *Integrated pest management: Innovation-development process*. Dordrecht: Springer, 2009. p. 83-87.
- REITZ, S. R. Biology and ecology of the western flower thrips (Thysanoptera: Thripidae): The making of a pest. *Florida Entomologist*, v. 92, n. 1, p. 7-13, 2009.

RILEY, D. G. *et al.* Thrips vectors of tospoviruses. *Journal of Integrated Pest Management*, v. 2, n. 1, p. 11-110, 2011.

SAFRATIVA. Tripes: fique livre dessas pragas de uma vez por todas. [2022]. Disponível em: <https://safrativa.com.br/tripes/> . Acesso em: 01 jul. 2025.

SALAS, Fernando Javier Sanhueza. *Cultura da batata: pragas e doenças*. São Paulo: Instituto Biológico, 2017 p. 22-24.