

Embedded neural networks for identifying *Spodoptera frugiperda* in corn plantations

Bianca Soares¹, Wendell Silva¹, Gabriela Ponciano¹, Bruna Stefanie¹,
Valentine Almeida², Patrik Pastori³, Deborah Magalhães⁴, Atslands Rocha¹

¹Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará

²Laboratório de Entomologia Aplicada, Universidade Federal do Ceará, Brasil

³Pós-Graduação em Agronomia/Fitotecnia, Universidade Federal da Grande Dourados

⁴Universidade da Integração Internacional da Lusofonia Afro-Brasileira (Unilab – CE)

{biancasoares, wendellsantos, gabriela.ponciano, brunastefanie}@alu.ufc.br

valentine.lalmeida@alu.ufc.br, patrikpastori@ufgd.edu.br

deborah.vm@unilab.edu.br, atslands@ufc.br

Abstract. *The Spodoptera frugiperda is one of the most important pests in global agriculture. Its monitoring typically requires visual inspection and manual counting of individuals. However, advances in computer vision, machine learning, and the Internet of Things offer ways to achieve fast and accurate monitoring. In this scenario, we investigated the use of lightweight convolutional neural networks and dense networks to extract relevant features from images of these insects captured in traps. The best extraction models, MobileNet and DenseNet201, were combined with MLP and achieved classification accuracies of 0.89 and 0.94, respectively, when deployed on a Raspberry Pi. Our results show that DenseNet201 offers higher accuracy than MobileNet. Nevertheless, MobileNet is more efficient in processing and has a shorter execution time. Therefore, it emerges as a viable alternative for identifying Spodoptera frugiperda in the field with computationally constrained devices. Finally, this work contributes directly to the automated and precise monitoring of crop pests.*

Resumo. *A Spodoptera frugiperda é uma das principais pragas da agricultura mundial. Seu monitoramento comumente requer inspeção visual e contagem manual de indivíduos. Avanços em visão computacional, aprendizado de máquina e internet das coisas oferecem meios para alcançar um monitoramento rápido e preciso. Neste cenário, nós investigamos o uso de redes neurais convolucionais leves e densas para extrair características relevantes de imagens desses insetos capturadas em armadilha. Os melhores modelos de extração, MobileNet e DenseNet201, foram aliados à MLP e alcançaram acurácia de classificação de 0,89 e 0,94, respectivamente, embarcados em Raspberry Pi. Os resultados mostram que, embora a DenseNet201 ofereça maior precisão, a MobileNet é mais eficiente em processamento e possui menor tempo de execução. Portanto, a MobileNet desponta como alternativa viável para identificação da Spodoptera frugiperda em campo com dispositivos com restrição computacional. Por fim, este trabalho contribui diretamente para o monitoramento automático e preciso de pragas em plantações.*

1. Introdução

O milho é um cereal fundamental para alimentação humana e animal, sendo a segunda *commodity* mais produzida no mundo, no qual o Brasil é o terceiro maior produtor mundial [FAO 2022]. A *Spodoptera frugiperda* (J.E. Smith) (Lepidoptera: Noctuidae) é um inseto-praga polífago que afeta o milho e causa danos em cerca de 353 espécies de diversas famílias botânicas [Montezano et al. 2018]. Portanto, é uma das principais pragas da agricultura mundial, tornando-se necessário seu constante monitoramento e controle.

O manejo integrado de pragas (MIP) consiste no uso de um conjunto de técnicas a fim de reduzir essa infestação ao nível de controle [Gallo et al. 2002b]. A forma de monitoramento mais comum da *S. frugiperda* em campo é feita por meio de armadilhas com algum tipo de atrativo, sendo utilizado, nesse caso, um feromônio sexual específico da espécie. Essas armadilhas são projetadas para realizar a captura de indivíduos adultos, geralmente machos no qual a identificação pode ser realizada através da visualização de padrões de cores em suas asas [Kenis et al. 2022]. Contudo, esse tipo de monitoramento requer uma averiguação visual no local que a armadilha foi posta, no qual é realizada a contagem manual do número de indivíduos capturados [Gallo et al. 2002a].

Pesquisas em Visão Computacional aliada ao Aprendizado de Máquina oferecem soluções promissoras para a detecção automática de insetos que atacam plantações [Karunathilake et al. 2023]. Ainda, os trabalhos de [Hassan et al. 2023] e [Azfar et al. 2023] abordam o uso de processamento de imagens combinado com a IoT (*Internet of Things*) para monitorar plantações de arroz e algodão, respectivamente.

Os modelos de aprendizado de máquina podem ser embarcados em dispositivos de baixo custo, oferecendo uma solução acessível para o monitoramento em campo [Ghazal et al. 2024]. No entanto, devido às limitações de hardware (processamento e memória), os modelos precisam ser otimizados para funcionar de forma eficiente em tempo real. Isso exige o uso de redes neurais mais leves e técnicas de compressão de modelos para minimizar a demanda de recursos [Muhammed et al. 2024]. Além disso, a variabilidade nas condições ambientais, como iluminação e umidade, pode impactar a precisão dos modelos [Hu et al. 2024]. Para superar esses desafios, é essencial combinar esses algoritmos com estratégias eficazes de pré-processamento de imagens e realizar um ajuste fino dos modelos, garantindo operação confiável em diferentes condições.

Neste trabalho, investigamos modelos que combinam características extraídas por Redes Neurais Convolucionais (CNNs) com algoritmos de classificação para identificar a *Spodoptera frugiperda*, usando imagens adquiridas por meio de armadilhas. Ainda, investigamos o desempenho da melhor rede densa e rede leve aliadas ao classificador *Multi Layer Perceptron* (MLP) embarcadas no Raspberry Pi. Deste modo, nossa proposta oferece uma alternativa flexível entre eficiência de recursos e precisão, contribuindo de forma significativa para o monitoramento da *Spodoptera frugiperda* e contribuindo para intervenções mais rápidas e eficazes no campo.

2. Trabalhos Relacionados

A Tabela 1 resume um comparativo dos trabalhos relacionados, considerando três aspectos: o problema abordado, o modelo utilizado e as métricas de avaliação empregadas.

Em [Obasekore et al. 2023] foi feita a detecção de larvas da *Spodoptera frugi-*

perda utilizando um base composta por 862 imagens aliadas à modelos de aprendizado profundo. Para a detecção de insetos, a Faster R-CNN apresentou o melhor desempenho, com uma acurácia de 0,99 e precisão média (mAP) de 0,84, demonstrando alta viabilidade para implantação em um cenário real.

[Junior and Rieder 2020] aborda o uso de técnicas de visão computacional para automatizar a contagem e identificação de pequenos insetos. Eles empregaram técnicas para a eliminação de ruídos, como a aplicação do filtro *GaussianBlur*, em uma base composta por 167 imagens, e usaram a rede Mask R-CNN para contagem e identificação dos insetos, alcançando uma mAP de 0,60 sugerindo a viabilidade da automação para melhorar a eficiência na identificação e contagem de insetos em contextos agrícolas.

No trabalho de [Mira et al. 2024], foram investigados vários algoritmos clássicos para detecção da *Bactrocera oleae* em dispositivos de borda, utilizando 3043 imagens capturadas em condições diversas. A combinação do algoritmo de Análise de componente principal (PCA) com o *Random Forest*, alcançou uma acurácia de 0,99 ao ser implantado em um Raspberry Pi 4, utilizando menos recursos em comparação com outros modelos, demonstrando a eficiência de algoritmos menos complexos no cenário embarcado.

[Kathole et al. 2023] desenvolveram uma arquitetura CNN (CNLTSM) que usa as características extraídas da YOLOv3 como descritor, para identificar insetos presentes no campo, utilizando a base de dados pública IP102¹. Para otimização é utilizado o algoritmo AHBA (*Adaptive Honey Badger Algorithm*), que alcançou 0,90 nas métricas de avaliação, apresentando um desempenho melhor comparado aos outros métodos.

[Guo et al. 2024] propõe um classificador de imagens de pragas em um mundo aberto, utilizando uma arquitetura leve e otimizada baseada na ResNet8, treinada com a função de perda de entropia cruzada escalada (NT-Xent). O método obteve uma acurácia de 0,84, sendo uma solução viável para o reconhecimento de pragas de mundo aberto.

Tabela 1. Comparação dos trabalhos relacionados

Trabalho	Problema	Modelo	Métricas
[Obasekore et al. 2023]	Detecção de larvas da <i>Spodoptera frugiperda</i>	Faster R-CNN	Acurácia 0,99 mAP 0,84
[Junior and Rieder 2020]	Contagem e identificação de pequenos insetos	Mask R-CNN	mAP 0,60
[Mira et al. 2024]	Detecção da <i>Bactrocera oleae</i>	PCA + RF	Acurácia 0,99 Acurácia 0,90
[Kathole et al. 2023]	Identificação de insetos no campo	YOLOv3 + AHBA-CNLTSM	Precisão 0,90 Especificidade 0,90 F1-score 0,90
[Guo et al. 2024]	Identificação de pragas em mundo aberto	ResNet8 + NT-Xent	Acurácia 0,84

3. Metodologia

Os experimentos de classificação utilizam uma metodologia (Figura 1) composta por seis etapas, na qual os resultados da etapa anterior são utilizados pela próxima etapa.

A primeira etapa consistiu na aquisição das imagens, na qual os dados são coletados em campo por meio de armadilhas. Na segunda etapa, o processamento das imagens

¹<https://github.com/xpwu95/IP102>

inclui operações como redimensionamento e normalização, assegurando uniformidade nas propriedades das imagens e preparando-as para a análise subsequente. Na terceira etapa, características são extraídas das imagens utilizando redes neurais convolucionais (CNNs) pré-treinadas, que capturam padrões e características visuais complexas. A quarta etapa envolve a classificação das imagens com base nas características extraídas, utilizando modelos de aprendizado de máquina para categorizar as imagens em duas classes: com e sem inseto. Na quinta etapa, diversas métricas de desempenho são empregadas para avaliar a eficácia do modelo de classificação. Por fim, na sexta etapa, os melhores modelos são embarcados em um dispositivo Raspberry Pi, visando avaliar sua performance em um ambiente com processamento e recursos de hardware limitados.



Figura 1. Metodologia para a classificação da *Spodoptera frugiperda*

3.1. Aquisição das imagens

A aquisição das imagens no estudo foi realizada através de uma armadilha inteligente equipada com uma câmera integrada a um Raspberry Pi, apresentado em [Silva et al. 2024]. A armadilha foi projetada para capturar imagens de indivíduos adultos da espécie *Spodoptera frugiperda* no campo. As imagens foram transmitidas via Wi-Fi para um servidor. Para validar a metodologia de coleta de imagens, testes de campo foram conduzidos em uma plantação de milho afetada pela *S. frugiperda*. A armadilha inteligente capturou imagens no período de 28/11/2023 a 30/11/2023, como apresentado na Tabela 2, resultando em uma base de dados de mais de 5000 fotos. Essas imagens foram posteriormente processadas e analisadas para realizar a identificação das pragas, empregando métodos de aprendizado de máquina.

Tabela 2. Distribuição das imagens por data

Data	Imagens com inseto	Imagens sem inseto	Total
28/11/2023	84	1803	1887
29/11/2023	3814	1900	5714
30/11/2023	1549	557	2106

3.2. Pré-processamento

No pré-processamento, as imagens foram normalizadas no intervalo [0, 1]. Desse modo, os valores dos pixels estão na mesma escala, evitando que valores mais altos dominem

o processo de treinamento e permitindo uma aprendizagem mais estável e eficiente dos modelos [Nixon and Aguado 2020]. Após a normalização, as imagens foram redimensionadas para o tamanho 224×224 devido a compatibilidade com arquiteturas CNN populares, como a MobileNet. O redimensionamento das imagens também permite que as CNNs aproveitem ao máximo suas camadas convolucionais e de *pooling* para extrair características significativas dos dados [Bechar et al. 2019].

3.3. Extração de características

O uso de arquiteturas CNNs pré-treinadas como descritores em problemas de classificação é uma técnica amplamente adotada na literatura [Palei et al. 2023] [Yodrot et al. 2024]. Nesse contexto, as camadas convolucionais das redes são empregadas para extrair características das imagens, que são usadas como entradas para classificadores mais simples, como SVM e MLP. Este método aproveita o poder de generalização das CNNs pré-treinadas, proporcionando uma base sólida de características visuais robustas que melhoram a precisão dos classificadores subsequentes. Além disso, essa abordagem é particularmente vantajosa quando se trabalha com conjuntos de dados menores [Ngugi et al. 2024].

As CNNs pré-treinadas podem ser divididas em duas categorias principais: redes leves e redes densas. Redes leves, como MobileNet, são projetadas para serem eficientes em termos de recursos computacionais, com menos parâmetros e menor demanda por processamento e memória. Estas redes são ideais para implementação em dispositivos com recursos limitados, como smartphones e sistemas embarcados [Yu et al. 2020]. Em contraste, redes densas, como a DenseNet, são mais complexas e profundas, com um grande número de camadas e parâmetros, oferecendo maior capacidade de aprendizado e melhor desempenho em tarefas complexas de visão computacional [Jiang et al. 2024].

Neste trabalho, foram extraídas características de seis arquiteturas CNNs pré-treinadas com a base de dados ImageNet [Russakovsky et al. 2015], utilizando a biblioteca Keras como *backend* da ferramenta TensorFlow. As redes leves utilizadas incluíram a MobileNet, EfficientNetB0 e NasNetMobile. As arquiteturas densas, são representadas pela DenseNet201, ResNet50 e VGG16. Para todos os modelos, as características foram obtidas da última camada de *pooling*, desconsiderando as camadas totalmente conectadas. Posteriormente, aplicou-se o método de média global de *pooling* ao mapa de características resultante, reduzindo o descritor para a profundidade da última camada de *pooling*. A entrada foi padronizada para todas as arquiteturas, mantendo dimensões iguais de $224 \times 224 \times 3$. A Tabela 3 apresenta as informações detalhadas sobre cada descritor.

Tabela 3. Propriedades dos descritores

Categoria	CNN	Parâmetros	Dimensão
Leve	EfficienteNetB0	5.3M	1280
	MobileNet	4.3M	1024
	NasNetMobile	5.3M	1056
Densa	DenseNet201	20.2M	1920
	ResNet50	25.6M	2048
	VGG16	138.4M	512

3.4. Classificação

Nessa etapa foram adotados quatro algoritmos de classificação: LightGBM (LGBM), Random Forest (RF), Support Vector Machine (SVM) e Multi-Layer Perceptron (MLP). O LGBM utiliza a técnica de *boosting*, onde múltiplas árvores de decisão são treinadas sequencialmente, com cada árvore corrigindo os erros da anterior. O RF também se baseia na utilização de múltiplas árvores de decisão e é menos suscetível ao sobreajuste. O SVM encontra o hiperplano que melhor separa as classes no espaço de características, maximizando a margem entre elas. Por fim, o MLP é um modelo de rede neural capaz de aprender representações complexas dos dados através de suas camadas ocultas.

As imagens foram divididas em 80% (7665) para o treinamento e 20% (1942) para a avaliação do desempenho da classificação. Os parâmetros dos classificadores foram definidos por meio do otimizador de busca em grade (*Grid Search*), garantindo que cada modelo fosse ajustado para obter o melhor desempenho possível. A Tabela 4 exibe o intervalo definido para cada parâmetro com os valores escolhidos marcados em negrito.

Tabela 4. Configuração dos hiperparâmetros dos classificadores otimizada por meio do algoritmo Grid Search.

Classificadores	Intervalo dos Parâmetros
LGBM/RF	estimators: [200, 300, 400 , 500 , 700]
SVM	kernel: [linear , rbf, poly , sigmoid], C: [0.1 , 0.5, 0.7 , 1.0, 1.5]
MLP	função de ativação: [relu , identity, tanh], solucionador: [adam , lbfgs, sgd], n° de épocas: [20, 30 , 50, 80]

3.5. Validação

Foram usadas as métricas acurácia, índice Kappa, precisão, revocação e F1-score para avaliar os resultados de classificação. Essas métricas são essenciais para a análise e aprimoramento dos modelos, permitindo uma compreensão detalhada de seu desempenho.

3.6. Implantação dos modelos em dispositivo com restrição de recursos

Os melhores modelos das categorias *redes leves* e *redes densas* foram embarcados em um Raspberry Pi zero 2w, uma das versões com menor poder computacional, escolhido por seu tamanho adequado à armadilha utilizada no campo. Este dispositivo possui o sistema operacional Raspberry Pi OS, um processador RP3A0 Quad-Core de 64 bits e 1 GHz, LAN sem fio 802.11 b/g/n (2,4 GHz), Bluetooth 4.2 e Bluetooth Low Energy (BLE) e com 512 MB de SDRAM, uma forma de memória semicondutora que pode ser executada em velocidades mais rápidas do que as de computadores convencionais. Para adaptar esses modelos às limitações de hardware do dispositivo, eles foram convertidos para o formato TensorFlow Lite, uma versão otimizada para dispositivos com recursos computacionais restritos. Durante a avaliação dos modelos, foram considerados diversos aspectos críticos, incluindo tempo de execução, consumo de energia, uso de memória e temperatura do dispositivo. Também foi capturada a média exponencial ponderada do número de processos em execução ou aguardando para serem executados durante períodos

de 1, 5 e 15 minutos. A fórmula é :

$$\text{Load}_t = \alpha \cdot \text{Load}_{\text{current}} + (1 - \alpha) \cdot \text{Load}_{t-1}$$

onde Load_t é a média de carga no tempo t , $\text{Load}_{\text{current}}$ é a carga atual (o número de processos em execução ou aguardando no momento atual), Load_{t-1} é a média de carga no tempo anterior ($t - 1$) e α é o fator de suavização, com observações recentes recebendo maior peso nos períodos de um minuto e menor peso nos períodos de quinze minutos. Os valores de carga (Load de 1 minuto, Load de 5 minutos e Load de 15 minutos) ajudam a avaliar a performance e a carga do sistema ao longo do tempo. Valores abaixo de 1 indicam que o sistema está subutilizado, valores próximos de 1 indicam plena utilização, e valores acima de 1 indicam sobrecarga, com processos aguardando execução.

4. Experimentos e Resultados

4.1. Experimentos de classificação

A Tabela 5 apresenta os resultados de classificação obtidos com as características extraídas das CNNs leves, que demonstraram que as arquiteturas com menos de 10M de parâmetros podem ser relevantes para o problema, facilitando a implantação em dispositivos IoT. A MobileNet teve o melhor desempenho em todos os classificadores, destacando-se o MLP com o melhor resultado, sendo 0,96 de acurácia, 0,86 de kappa e 0,94 de precisão, revocação e F1-score. O valor do kappa ser abaixo das outras métricas deve-se ao desbalanceamento das classes. O pior desempenho foi obtido com a EfficientNetB0 combinada com SVM e MLP, com resultados variando entre 0,57 e 0,72. Apesar de ter um desempenho inferior ao da MobileNet, a NasNetMobile apresentou bons resultados, indicando que a combinação com o MLP pode ser uma solução promissora.

Tabela 5. Classificação das características extraídas das arquiteturas leves

Descritor	Clf	Acurácia	Kappa	Precisão	Revocação	F1-score
EfficientNetB0	LGBM	0,8928	0,7811	0,9072	0,9056	0,9064
	RF	0,8640	0,7217	0,8780	0,8858	0,8819
	SVM	0,5731	0,6854	0,7231	0,7345	0,7286
	MLP	0,5736	0,5856	0,6731	0,6631	0,6794
MobileNet	LGBM	0,9258	0,8480	0,9276	0,9443	0,9359
	RF	0,8960	0,7852	0,8863	0,9389	0,9119
	SVM	0,9300	0,8571	0,9421	0,9353	0,9387
	MLP	0,9620	0,8616	0,9496	0,9408	0,9401
NasNetMobile	LGBM	0,8980	0,7905	0,8989	0,9263	0,9124
	RF	0,8435	0,6744	0,8297	0,9146	0,8701
	SVM	0,8919	0,7791	0,9064	0,9048	0,9056
	MLP	0,8970	0,7894	0,9087	0,9119	0,9103

Os resultados de classificação utilizando arquiteturas densas são apresentados na Tabela 6 e demonstram um desempenho relevante. A combinação da DenseNet201 com o classificador MLP alcançou 0,95 em acurácia, precisão, revocação e F1-score, e 0,90 no índice Kappa, superando os resultados obtidos com a MobileNet. Em contraste, a

arquitetura VGG16 com o SVM obteve pior desempenho, com métricas variando entre 0,26 e 0,93, indicando inconsistências na classificação e uma menor capacidade de generalização. Esses resultados destacam a eficácia da DenseNet201 na classificação da *Spodoptera frugiperda*. A alta consistência nas métricas de avaliação ressalta a capacidade do modelo de capturar com precisão as características distintivas das imagens.

Tabela 6. Classificação das características extraídas das arquiteturas densas

Descritor	Clf	Acurácia	Kappa	Precisão	Revocação	F1-score
DenseNet201	LGBM	0,9470	0,8915	0,9501	0,9578	0,9539
	RF	0,9125	0,8201	0,9104	0,9398	0,9248
	SVM	0,8939	0,7831	0,9060	0,9093	0,9076
	MLP	0,9531	0,9027	0,9530	0,9551	0,9542
ResNet50	LGBM	0,9135	0,8230	0,9215	0,9281	0,9248
	RF	0,8883	0,7697	0,8836	0,9272	0,9049
	SVM	0,7353	0,4458	0,7360	0,8392	0,7842
	MLP	0,7379	0,4782	0,8213	0,6936	0,7521
VGG16	LGBM	0,9434	0,8839	0,9426	0,9596	0,9510
	RF	0,9145	0,8245	0,9143	0,9389	0,9264
	SVM	0,6663	0,2617	0,6445	0,9317	0,7619
	MLP	0,8965	0,7847	0,8719	0,9505	0,9141

Dessa forma, o classificador MLP combinado com as redes MobileNet e DenseNet201 destacaram-se pelos melhores resultados obtidos. Apesar de ser uma rede simples, a MobileNet alcançou um resultado próximo ao da DenseNet201. Para avaliar essas arquiteturas em um cenário com limitação de hardware, implementamos os modelos em um dispositivo Raspberry Pi considerando a precisão do modelo e o custo computacional.

4.2. Modelos embarcados

No campo, a internet é instável, tornando mais eficiente o envio dos resultados em vez de imagens. Para lidar com essa limitação, os melhores modelos foram convertidos para o formato TensorFlow Lite (.tflite) e reavaliados com o mesmo conjunto de teste dos experimentos anteriores. Além disso, o tempo de execução foi analisado, considerando a média, o menor e o maior tempo, como mostrado na Tabela 7. Após a conversão, observou-se uma perda de precisão após a conversão dos modelos para o TensorFlow Lite, exceto pela revocação da MobileNet+MLP, que aumentou de 0,94 para 0,95. Essa perda de precisão se deve ao processo de conversão e à arquitetura simplificada da rede. Já o modelo DenseNet201+MLP mostrou-se mais estável, com uma leve melhora na revocação e variações mínimas nas demais métricas, destacando a robustez da sua arquitetura densa em comparação com a MobileNet.

Ao analisar o tempo de execução, percebe-se uma diferença significativa no desempenho entre os dois modelos. A MobileNet+MLP demonstrou tempos de execução médios mais curtos, com uma média de 0,3637 segundos e maior tempo de 6,5682 segundos. Em contraste, a DenseNet201+MLP apresentou uma média de tempo de execução de 2,3218 segundos e maior tempo de 12,5180 segundos. O tempo total de execução acumulado para a MobileNet+MLP foi de 706,45 segundos, enquanto para a Dense-

Net201+MLP foi de 4508,96 segundos. Esses dados sugerem que, embora a DenseNet201+MLP possa oferecer uma maior precisão em determinadas métricas, a MobileNet+MLP é mais eficiente em termos de tempo de processamento, crucial em aplicações onde a velocidade é essencial.

Tabela 7. Resultados para os modelos embarcados

Métricas	MobileNet+MLP	DenseNet201+MLP
Acurácia	0,8923	0,9479
Kappa	0,7784	0,8940
Precisão	0,8686	0,9427
Revocação	0,9522	0,9660
F1-score	0,9085	0,9542
Média tempo de execução (s)	0,3637	2,3218
Menor tempo de execução (s)	0,3282	1,9236
Maior tempo de execução (s)	6,5682	12,5180
Total tempo de execução (s)	706,45	4508,96

A Tabela 8 apresenta o tempo de execução, consumo de energia, uso de memória e temperatura do dispositivo capturados no momento de execução dos modelos. As Figuras 2 e 3 representam as cargas para os períodos de 1, 5 e 15 minutos durante a execução dos modelos MobileNet+MLP (de 15:55 a 16:10) e DenseNet201+MLP (de 22:40 a 23:50).

Tabela 8. Informações do Raspberry Pi enquanto executava os modelos.

Métricas	MobileNet+MLP	DenseNet201+MLP
Temperatura (°C)	64,242	67,072
Total Memory (MB)	419,336	419,336
Available Memory (MB)	268,828	294,127
Free Memory (MB)	184,861	148,621
SWAP Used (MB)	109,197	218,250
Potência (mW)	2,962	3,15

Observou-se que o modelo DenseNet201+MLP gerou uma temperatura mais alta no dispositivo (67,072 °C) em comparação com o MobileNet+MLP (64,242 °C), sugerindo maior demanda de processamento e aquecimento. Ambos os modelos tinham uma memória total disponível de 419,336 MB. O DenseNet201+MLP deixou mais memória disponível para outros processos (294,127 MB) em comparação ao MobileNet+MLP (268,828 MB), mas teve menos memória livre (148,621 MB) do que o MobileNet+MLP (184,861 MB). Indicando maior consumo de memória pelo DenseNet201+MLP. DenseNet201+MLP com 145,51 MB e o MobileNet+MLP com 83,97 MB.

O uso de SWAP foi significativamente maior para o DenseNet201+MLP (218,250 MB) do que para o MobileNet+MLP (109,197 MB), indicando um maior consumo de recursos além da RAM física. O consumo de potência foi ligeiramente maior para o DenseNet201+MLP (3,15 mW) em comparação ao MobileNet+MLP (2,962 mW), corroborando a maior demanda de processamento e energia. Analisando as cargas de sistema

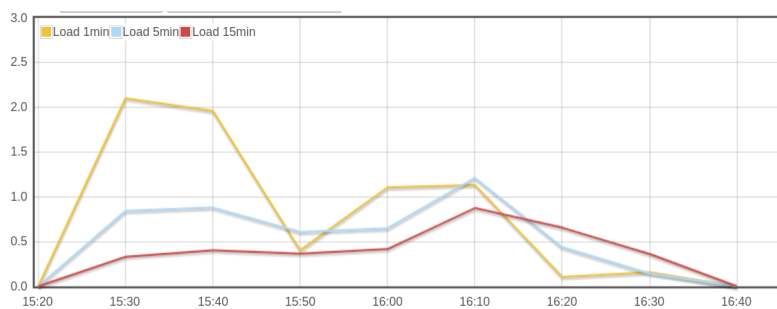


Figura 2. MobileNet+MLP: Loads de 1, 5 e 15 minutos.

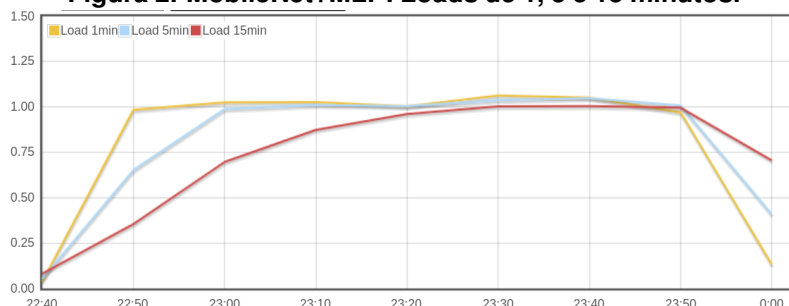


Figura 3. DenseNet201+MLP: Loads de 1, 5 e 15 minutos.

em períodos de 1, 5 e 15 minutos, os valores foram 1,134, 1,121 e 0,874 para o MobileNet+MLP e 1,186, 1,083 e 1,017 para o DenseNet201+MLP. As cargas mais altas para o DenseNet201+MLP indicam um maior uso de CPU, com exceção da carga de 5 minutos, onde o MobileNet+MLP teve uma carga ligeiramente maior.

5. Considerações Finais

Neste trabalho, investigou-se a combinação de características extraídas de CNNs pré-treinadas com algoritmos de classificação para identificar a presença da *Spodoptera frugiperda* em imagens. A principal contribuição deste estudo consiste em investigar o uso de redes leves e densas para descrever a presença da *Spodoptera frugiperda* a fim de auxiliar produtores na detecção precoce e no controle eficiente dessa praga agrícola.

Foram desenvolvidos dois modelos que combinam as arquiteturas MobileNet e DenseNet201 com o classificador MLP, alcançando uma acurácia de 0,96 e 0,95 respectivamente. Esses modelos foram implementados em um dispositivo Raspberry Pi para testar sua viabilidade em ambientes embarcados. O modelo baseado na DenseNet201, apesar de consumir mais recursos, utilizou de forma eficiente o sistema computacional, apresentando poucos picos de sobrecarga, tornando-se uma opção viável em cenários onde as limitações de hardware não são severas. Já para ambientes com restrições de recursos mais rigorosas, o modelo baseado na MobileNet é mais adequado devido à sua eficiência em tempo de execução e baixo uso de recursos computacionais.

Futuramente, planeja-se ajustar outros parâmetros do modelo melhorar a precisão no ambiente embarcado e explorar sua capacidade de generalização do modelo para diferentes conjuntos de dados, avaliando sua robustez e aplicabilidade em novos cenários. Além dessas melhorias, pretende-se expandir a aplicação do modelo para outros dispositivos embarcados, analisando o desempenho em termos de consumo de energia e eficiência.

6. Agradecimentos

Este trabalho recebeu apoio da Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (FUNCAP) através do projeto MLC-0191-00164.01.00/22.

Referências

- Azfar, S., Nadeem, A., Ahsan, K., Mehmood, A., Almoamari, H., and Alqahtany, S. S. (2023). Iot-based cotton plant pest detection and smart-response system. *Applied Sciences*, 13(3):1851.
- Bechar, M. E. A., Settouti, N., Daho, M. E. H., Adel, M., and Chikh, M. A. (2019). Influence of normalization and color features on super-pixel classification: application to cytological image segmentation. *Australasian physical & engineering sciences in medicine*, 42:427–441.
- FAO (2022). Faostat: Crops and livestock products. Accessed: 2024-07-30.
- Gallo, D., Nakano, O., Silveira Neto, S., Carvalho, R., Baptista, C., Berti Filho, E., Parra, J., Zucchi, R., Alves, S., Vendramim, J., et al. (2002a). Agricultural entomology= entomologia agrícola. fealq, piracicaba, sp, brazil.
- Gallo, D., Nakano, O., Silveira Neto, S. S., Carvalho, R. P. L., Batista, G. C., Filho, E. B., P., P. J. R., Zucchi, R. A., Alves, S. B., Vendramim, J. D., Marchini, L. C., Lopes, J. R. S., and Omoto, C. (2002b). *Entomologia agrícola*. FEALQ, Piracicaba.
- Ghazal, S., Munir, A., and Qureshi, W. S. (2024). Computer vision in smart agriculture and precision farming: Techniques and applications. *Artificial Int. in Agriculture*.
- Guo, Q., Wang, C., Xiao, D., and Huang, Q. (2024). A lightweight open-world pest image classifier using resnet8-based matching network and nt-xent loss function. *Expert Systems with Applications*, 237:121395.
- Hassan, S. I., Alam, M. M., Illahi, U., and Suud, M. M. (2023). A new deep learning-based technique for rice pest detection using remote sensing. *PeerJ Computer Science*, 9:e1167.
- Hu, T., Zhang, X., Khanal, S., Wilson, R., Leng, G., Toman, E. M., Wang, X., Li, Y., and Zhao, K. (2024). Climate change impacts on crop yields: A review of empirical findings, statistical crop models, and machine learning methods. *Environmental Modelling & Software*, page 106119.
- Jiang, S., Luo, B., Jiang, H., Zhou, Z., and Sun, S. (2024). Research on dense object detection methods in congested environments of urban streets and roads based on dcyolo. *Scientific Reports*, 14(1):1127.
- Junior, T. D. C. and Rieder, R. (2020). Uma implementação baseada em mask r-cnn para detecção de insetos em imagens digitais. *Rev. Bras. de Entomologia*, 64(2):149–157.
- Karunathilake, E., Le, A. T., Heo, S., Chung, Y. S., and Mansoor, S. (2023). The path to smart farming: Innovations and opportunities in precision agriculture. *Agriculture*, 13(8):1593.
- Kathole, A. B., Katti, J., Lonare, S., and Dharmale, G. (2023). Identify and classify pests in the agricultural sector using metaheuristics deep learning approach. *Franklin Open*, 3(1):100024.

- Kenis, M., Benelli, G., Biondi, A., Calatayud, P.-A., Day, R., Desneux, N., Harrison, R. D., Kriticos, D., Rwomushana, I., van den Berg, J., et al. (2022). Invasiveness, biology, ecology, and management of the fall armyworm, *spodoptera frugiperda*. *Entomologia Generalis*.
- Mira, J. L., Gómez, S., del Río, A., Díaz, F., Lozano, M., and García, L. (2024). Benchmarking of computer vision methods for energy-efficient high-accuracy olive fly detection on edge devices. *Multimedia Tools and Applications*, 83(1):1203–1220.
- Montezano, D. G., Sosa-Gómez, D., Specht, A., Roque-Specht, V. F., Sousa-Silva, J. C., Paula-Moraes, S. d., Peterson, J. A., and Hunt, T. (2018). Host plants of *s. frugiperda* (lepidoptera: Noctuidae) in the americas. *African entomology*, 26(2):286–300.
- Muhammed, D., Ahvar, E., Ahvar, S., Trocan, M., Montpetit, M.-J., and Ehsani, R. (2024). Artificial Intelligence of Things (AIoT) for smart agriculture: A review of architectures, technologies and solutions. *Journal of Network and Computer Applications*, 228:103905.
- Ngugi, H. N., Ezugwu, A. E., Akinyelu, A. A., and Abualigah, L. (2024). Revolutionizing crop disease detection with computational deep learning: a comprehensive review. *Environmental Monitoring and Assessment*, 196(3):302.
- Nixon, M. S. and Aguado, A. S. (2020). Image processing. In *Feature Extraction and Image Processing for Computer Vision*, pages 83–139. Elsevier.
- Obasekore, H., Fanni, M., Ahmed, S. M., Parque, V., and Kang, B.-Y. (2023). Agricultural robot-centered recognition of early-developmental pest stage based on deep learning: A case study on fall armyworm (*spodoptera frugiperda*). *Sensors*, 23(6):3147.
- Palei, S., Lenka, R. K., Nayak, S. S., Mohanty, R., Jena, B., and Saxena, S. (2023). Precision Agriculture: ML and DL-Based Detection and Classification of Agricultural Pests. In *2023 2nd International Conference on Ambient Intelligence in Health Care (ICAIHC)*, pages 1–6, Bhubaneswar, India. IEEE.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252.
- Silva, W. d. S., Soares, B., Almeida, V. d. L., Viana, L., Pastori, P. L., Magalhaes, D. M., and da Rocha, A. R. (2024). Detecção da praga *spodoptera frugiperda* no cultivo de milho usando armadilhas inteligentes e visao computacional. In *Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais (WCAMA)*, pages 61–70. SBC.
- Yodrot, T., Sutacha, C., Orachon, T., Jangjongdee, N., and Boonyasuwan, S. (2024). A Study on the Potency of Hybrid Models: Detecting Diseases in Cucumber Leaves with Pre-trained CNNs and SVM. In *2024 12th International Electrical Engineering Congress (iEECON)*, pages 1–4, Pattaya, Thailand. IEEE.
- Yu, D., Xu, Q., Guo, H., Zhao, C., Lin, Y., and Li, D. (2020). An efficient and lightweight convolutional neural network for remote sensing image scene classification. *Sensors*, 20(7):1999.