Automatic Detection of Spots on Coffee Leaves using Deep Learning

Caio E. Dias
LIAREA and PGCAP
INPE
São José dos Campos, Brazil
caio.dias@inpe.br

Valdivino A. Santiago Júnior
LIAREA and PGCAP
INPE
São José dos Campos, Brazil
valdivino.santiago@inpe.br

Elcio E. Shiguemori
PGCAP
IEAv
São José dos Campos, Brazil
elcio@ieav.cta.br

***Abstract*—Coffee cultivation is an agricultural activity of great importance to the global economy, but it faces challenges due to diseases and pests that compromise crop productivity. Therefore, the automatic detection of spots on coffee leaves, which may indicate diseases and pests such as rust and leaf miner, is extremely relevant for increasing productivity in plantations. The use of deep learning (DL) models has become increasingly common for this problem, but robust evaluations are still necessary to understand the suitability of these models, as well as the importance of including both laboratory and field images. This study evaluates the performance, in terms of accuracy and computational cost, of seven DL-based object detectors considering two datasets of coffee leaves. Models such as YOLOv10-M and YOLOv10-X offer a good balance between accuracy and latency but still present high inference times for embedded applications, especially when compared to lighter models like YOLOv10-Nano. Among all the approaches analyzed, YOLOv10-Nano stands out as the most efficient option, offering a good balance between accuracy (0.332 mAP@50) and speed, with the lowest latency (8 ms) and a reduced training time (40 minutes). In addition, the Eigen-CAM technique, an explainable artificial intelligence (XAI) approach, was applied to the YOLOv10-Nano model with the aim of improving its interpretability, allowing for a better understanding of how the network focuses on different regions of the leaf during the detection process. The YOLOv10-Nano model has started to be deployed on mobile devices, aiming at future developments that will make this technology viable for small producers.**

***Index Terms*—Coffee, Object Detection, Leaves, Deep Learning, YOLO**

I. Introduction

Coffee production has a significant influence on the global economy, being the second most consumed beverage after water and a source of livelihood for millions of farmers in various regions [1]. As the world's largest producer and exporter, Brazil plays a central role in this market. In 2024, the gross revenue of Brazilian coffee farming was estimated at R$ 79.59 billion [2]. The sector comprises approximately 330,000 rural properties across 17 states, 78% of which are family farms [3]. However, coffee farming faces constant challenges, such as the impact of diseases and pests, which compromise productivity and grain quality, threatening the economic sustainability of plantations. Among the main phytosanitary threats are diseases that cause leaf spots on plants, impairing their photosynthetic capacity and resulting in premature leaf drop. Coffee leaf rust (*Hemileia vastatrix*), illustrated in Figure 1 (A), is one of the most devastating diseases, characterized by the appearance of orange lesions on the underside of leaves, which can lead to production losses of up to 30% if not controlled in time [4].

In addition to this, phoma leaf spot (*Phoma costarricensis*; Figure 1 (B)) and cercospora leaf spot (*Cercospora coffeicola*; Figure 1 (C)) also pose significant risks. The former causes dark lesions and necrosis in plant tissues, affecting branches and fruits, while the latter manifests as circular brown spots, compromising the plant's healthy development and reducing the quality of harvested beans [5].

Beyond the mentioned diseases, the coffee leaf miner (*Leucoptera coffeella*; Figure 1 (D)) is one of the most damaging pests in coffee farming, causing severe losses. Its larvae feed on the leaf parenchyma, creating galleries that reduce the plant's photosynthetic capacity. Severe infestations can lead to intense defoliation, resulting in productivity decline and plant weakening, making them more susceptible to other adversities. [6].

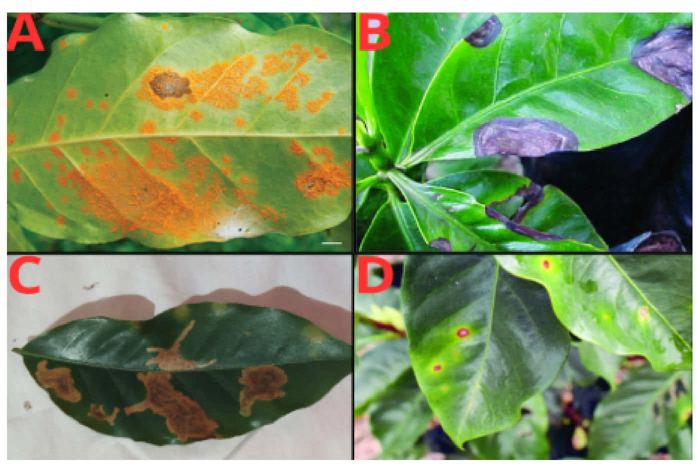

Fig. 1. Diseases and Pests Affecting Coffee Plantations. (A) Coffee Rust [7]. (B) Phoma [8]. (C) Coffee Leaf Miner [9]. (D) Cercosporiosis [10].

Given these challenges, efficient crop monitoring becomes essential to minimize impacts. Conventional inspection meth-

LIAREA - Laboratório de Inteligência ARtificial para Aplicações AeroEspaciais e Ambientais; PGCAP - Programa de Pós-Graduação em Computacção Aplicada; INPE - Instituto Nacional de Pesquisas Espaciais; IEAv - Instituto de Estudos Avançados.

ods, while commonly used, have limitations in terms of accuracy and response time [11]. This has driven the development of new approaches based on artificial intelligence (AI), e.g., machine learning (ML) and deep learning (DL), and computer vision, enabling the early detection of symptoms directly on plant leaves.

Previous studies have explored different DL approaches for detecting diseases in coffee leaves. Esgario et al. [12] used controlled-condition images for classification and quantification, while Visão et al. [13] evaluated multiple neural network architectures with real field images, focusing on coffee rust. Meanwhile, Araaf et al. [14] integrated YOLO models into an automated pruning system but did not consider alternative architectures.

Despite all these interesting initiatives already underway, three points can be mentioned as their limitations. Some of the studies are limited to the use of images in controlled conditions, with standardized lighting and positioning. They do not also use laboratory and field leaf images together to account for different situations. A second point is that, in this context, there is a need for more robust evaluations, considering several DL models, different performance perspectives (accuracy, computational cost) and different types of diseases and pests. Furthermore, initiatives to incorporate the model into mobile devices are scarce or even absent in this regard.

This study aims to address these issues and evaluates the performance, in terms of accuracy and computational cost, of seven DL-based object detectors using two coffee datasets composed of laboratory and field images. In the best-performing model, explainable artificial intelligence (XAI) techniques were applied to enhance the interpretability of the results. This best model has begun to be integrated into a mobile device as part of future efforts to make this technology accessible to small-scale farmers.

II. Related work

In the study by [12], the authors applied convolutional neural networks (CNNs) to identify and quantify diseases and pests in coffee leaves, exploring two approaches: a direct classification model and another based on semantic segmentation before classification. Direct classification of whole leaves achieved over 95% accuracy, while prior segmentation of symptoms increased accuracy to over 97%, with a determination coefficient close to 0.98 for severity estimation. However, the study used images captured under controlled conditions, which may compromise the model's applicability in the field, where environmental factors such as lighting and complex backgrounds can affect segmentation accuracy.

In [13], the researchers evaluated six neural network architectures (AlexNet, DenseNet, Inception, ResNet, SqueezeNet, and VGG) using the RoCoLe dataset, which consists of real images of coffee leaves collected in the field. The approach included binary classification to differentiate healthy from diseased leaves and multiclass classification to identify rust stages. ResNet achieved the best performance in binary classification, with 95.19% accuracy, while multiclass classification reached 78.03%, highlighting difficulties in distinguishing advanced disease stages. Despite the good results, the study focused exclusively on coffee rust and analyzed individual leaves, which may limit its application in scenarios where multiple leaves need to be evaluated simultaneously.

In [14], the combination of YOLOv5 and YOLOv8 networks was proposed to detect diseased leaves in images captured in the field, integrating the model into an automated pruning system to remove the most affected leaves. Using a dataset of 2,024 images collected on a farm in Indonesia, YOLOv8 achieved the best performance, reaching a mAP of 73.2% compared to 70.5% for YOLOv5. Implementation on a Jetson Nano enabled real-time detection at 10 FPS, demonstrating advances in automated crop management. However, the system was trained exclusively to detect coffee rust, without covering other diseases, and relied solely on YOLO models, without exploring other architectures that could offer better performance or complement its approach. Additionally, the processing rate may be a limiting factor for large-scale applications.

We highlight here the limitations of these studies and the key differences compared to our research. Previous approaches are restricted to a single source, either field or laboratory, whereas our study combines images from both to address different scenarios. Moreover, there is a need for more comprehensive evaluations, considering multiple DL models and various performance metrics, such as accuracy and computational cost. Thus, we aim to contribute to this advancement. Another important aspect is the deployment of models on mobile devices. To the best of our knowledge, none of the previous studies have explored this approach. Therefore, we integrated the best model identified in our experiments into a mobile device, expanding its potential applications.

III. Materials and Methods

A. Datasets

To ensure greater data diversity, this study used two image datasets. The first, Disease and Pest in Coffee Leaves [15], consists of 285 images of coffee leaves affected by rust and 257 with leaf miner. While it includes two types of diseases and pests—unlike previous studies that typically focus on a single issue (e.g., rust)—our main interest was the accurate localization of the lesions. The images were collected in a laboratory under controlled background and lighting conditions from coffee plantation leaves in Brazil. With a resolution of 4000×2250 pixels, they contain bounding boxes that highlight the affected areas of each disease. Figure 2 (A) presents an example.

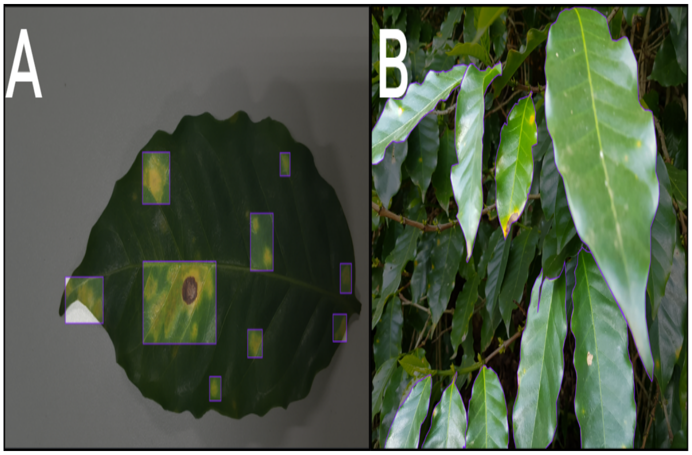

Fig. 2. Examples of images from the datasets used in this study. (A) Image from the *Disease and Pest in Coffee Leaves* dataset. (B) Image from the *BRACOT - A Brazilian Arabica Coffee Tree images dataset for instance segmentation of coffee leaves dataset.*

The second dataset, *BRACOT - A Brazilian Arabica Coffee Tree images dataset for instance segmentation of coffee leaves* [16], consists of images of coffee trees collected with a Galaxy S8 smartphone in the mountainous regions of Santa Maria and Marechal Floriano in the state of Espírito Santo, Brazil. Initially, this dataset included images and leaf segments affected by diseases. However, only the images were used, and the original segmentation was discarded. Bounding boxes were created around any signs of spots on the leaves. A total of 300 images were selected for the experiments. An example of the images contained in the dataset can be seen in Figure 2 (B).

The images were resized according to the standard input sizes of each model (see Section III-B). RT-DETR uses 480x480, while SSD MobileNet and EfficientDet work with 320x320. The YOLO variants (X, Nano, and M) use 640x640. We combined the images of both datasets and considered all the images in a unique set for the evaluation. Independent training, validation, and test sets were then created. Initially, 80% (673 images) of the images were randomly selected for training. Then, 10 images were reserved for testing, and the rest were used for validation (158 images).

B. DL models

Seven different approaches were considered for the object detection task, each bringing specific characteristics. Various architectures were evaluated, including RetinaNet, which introduced Focal Loss to mitigate the class imbalance, combining a backbone network like ResNet with Feature Pyramid Network (FPN) for feature extraction and an anchor-based detection head [17]. RT-DETR, an optimized version of DETR, reduces computational cost by integrating efficient convolutions and optimizing the self-attention mechanism of Transformers [18].

YOLO (You Only Look Once) treats object detection as a direct regression problem, enabling speed and accuracy by simultaneously predicting both bounding boxes and object classes in a single pass through the image [19]. Over time, the model evolved to meet different computational demands and application scenarios. It introduces variants that balance performance and efficiency according to the user's needs. YOLOv10-N is a compact version optimized for hardware-constrained devices such as embedded systems and mobile devices, offering a good compromise between lightness and accuracy. YOLOv10-M offers an intermediate configuration, providing a balance between speed and accuracy, suitable for real-time applications that require robust performance without compromising computational efficiency. YOLOv10-X is the most powerful variant, designed for scenarios that require high detection capability and fine object details, utilizing a deeper model with more parameters, which results in more intensive processing but also superior performance in identifying complex targets [20].

The MobileNet-SSD combines MobileNet as a feature extractor with SSD (Single Shot MultiBox Detector), utilizing depthwise separable convolutions to optimize computational efficiency while maintaining a good balance between accuracy and speed [21]. This approach makes the model particularly suitable for devices with hardware constraints, such as smartphones and embedded systems, ensuring fast processing without significantly compromising detection quality. On the other hand, EfficientDet, based on the EfficientNet architecture, employs *compound scaling* to balance network width, depth, and resolution, enhancing accuracy without substantially increasing computational demands [22]. Additionally, it incorporates BiFPN (*Bi-Directional Feature Pyramid Network*), which enables more efficient fusion of multiple feature levels, improving object detection across different scales with lower computational cost.

The models (RT-DETR, SSD MobileNet, EfficientDet, YOLO X, Nano, and M) were trained for 100 epochs on Google Colab using an NVIDIA T4 GPU. AdamW was chosen as the optimizer due to its improved generalization capability, as it decouples weight decay from gradient updates, reducing overfitting and enhancing training stability [23]. A low learning rate (0.00005) was selected to ensure smooth convergence, prevent gradient oscillations, and facilitate fine-tuning of pre-trained weights without compromising previously learned representations [24]. No early stopping was applied to guarantee a fair comparison of training time across all models.

IV. Explainable Artificial Intelligence

Explainable artificial intelligence (XAI) is a set of methods and techniques aimed at making artificial intelligence systems more transparent and understandable to humans. As ML models—especially complex ones such as deep neural networks—become increasingly present in real-world applications, so does the need to understand how these models make decisions. Many of these algorithms operate as "black boxes," producing predictions without providing clear explanations about the criteria or paths that led to the outcome [25], [26].

XAI emerges precisely to address this issue, seeking to develop approaches that reveal internal processes and the relationships between input and output in a comprehensible way. This explainability not only increases trust in AI models but also allows experts to identify failures, biases, inconsistent decisions, and ethical risks involved in the model's application.

A. Eigen-CAM

Eigen-CAM is an explanation technique that enables the interpretation of how neural network models make decisions based on images. Originally designed for classification tasks, it is also used in object detection, offering a way to visualize which regions of the image most influence the model's decision when identifying and locating objects. It generates activation maps that highlight the relevant areas used by the model during inference. Unlike other techniques that rely on gradient calculations to produce attention maps, it uses a decomposition based on eigenvalues and eigenvectors of the network's activations, capturing the main direction of variation of the responses from the last convolutional layer [27]. When applied to object detection models, it is possible to visualize whether the model is truly focusing on the correct regions when detecting a specific class or if it is making decisions based on irrelevant patterns.

Eigen-CAM was applied to the YOLOv10-Nano model with the aim of investigating which regions of the images had the greatest influence during the object detection process. In addition to enabling the visualization of the most relevant areas for the model's decisions, this application also served as a basis for analyzing the influence of the compact architecture's complexity on the model's attention. This made it possible to observe how the reduced capacity and depth of the network affect the focus of activations, providing a qualitative assessment of the model's interpretability and sensitivity in the face of structural limitations.

B. Metrics

Performance evaluation considered three main metrics: mAP@50, training time, and latency. mAP@50 (Mean Average Precision at 50% IoU) is a metric used to evaluate the precision of object detection models. It calculates the average precision for different classes of the model, considering a fixed threshold of 50% Intersection over Union (IoU). IoU measures the overlap between the bounding box predicted by the model and the real (ground truth) box, defined as the ratio between the intersection area and the union area of these boxes. Thus, mAP@50 indicates the average precision considering that a prediction is correct when the overlap between the predicted and real boxes reaches at least 50%. In this work, we focused solely on mAP@50 rather than higher IoU thresholds, since the primary goal was to ensure broader detection of leaf spots rather than highly precise localization. This is because even approximate detections can be valuable for early diagnosis and monitoring, where detecting a larger number of affected areas is more important than achieving exact bounding box alignment.

Latency represents the average time required for the model to process a single image and generate a prediction. It is a crucial factor, especially in real-time applications, such as video monitoring or detection on embedded devices. Latency is typically measured in milliseconds (ms) and can be influenced by various factors, such as the model complexity, the hardware used, and the efficiency of the inference code.

V. RESULTS

Table 1 presents a comparison of the different models used in this study. The models were evaluated based on three metrics as previously mentioned: one metric is related to the "accuracy"/precision (mAP@50), and the other two refer to the computational cost (inference latency, and training time).

Model	**mAP@50**	**Latency (ms)**	**Training time (h)**
RetinaNet	**0.38**	56	3:15
RT-DETR	**0.38**	101.93	3:00
YoloV10-M	0.37	41.2	3:30
YoloV10-X	0.36	42.5	5:51
YoloV10-Nano	0.332	**8**	0:40
Mobilenet-SSD	0.26	58.96	0:32
EfficientDet-SSD	0.23	275.08	**0:31**

TABLE I
COMPARISON OF THE PERFORMANCE OF DIFFERENT OBJECT DETECTION MODELS. BEST VALUES ARE IN BOLD.

RT-DETR (see Figure 4 (A)) and RetinaNet (see Figure 3 (A)) exhibit the highest precision among the evaluated models, achieving 0.38 mAP@50. However, this high precision comes with increased inference latency, which may compromise their viability for real-time or embedded applications. The RT-DETR model, for instance, exceeds 100 ms of latency, making it less suitable for embedded systems or devices with computational constraints. Although RT-DETR was designed to be a faster and lighter model, in this study, it did not achieve the expected performance. Figures 4 (C) (RT-DETR) and c (C) (RetinaNet) highlight examples of correct detections, where the models successfully identified leaf spots with precision.

Beyond the latency issue, an analysis of the test set images reveals a limitation of these models. Despite achieving the best precision in terms of mAP@50, they frequently misidentify background elements, such as branches and shadows, as leaf spots. Examples of these incorrect detections can be seen in Figures 4 (B) (RT-DETR) and 3 (B) (RetinaNet). This problem may be related to visual similarities between spots and background textures, dataset balance, or how the models extract features and fuse information at different scales. If the model fails to adequately separate these visually ambiguous regions, false positives may occur, affecting detection reliability.

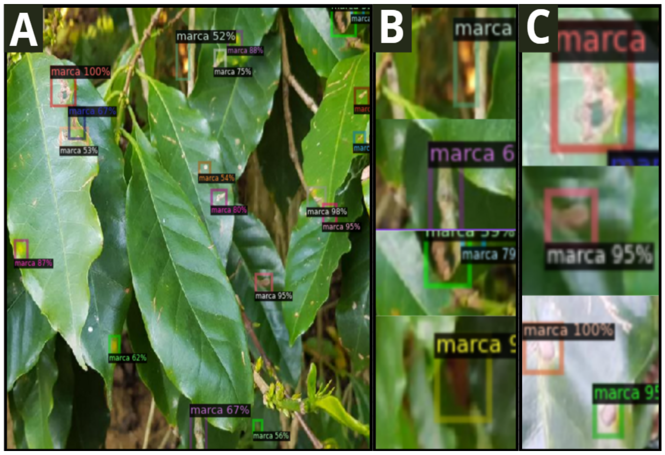

Fig. 3. Example of RetinaNet model output. (A) Detecting spots on coffee leaves in the field. (B) Examples of incorrect detection performed by the model. (C) Examples of correct detection performed by the model.

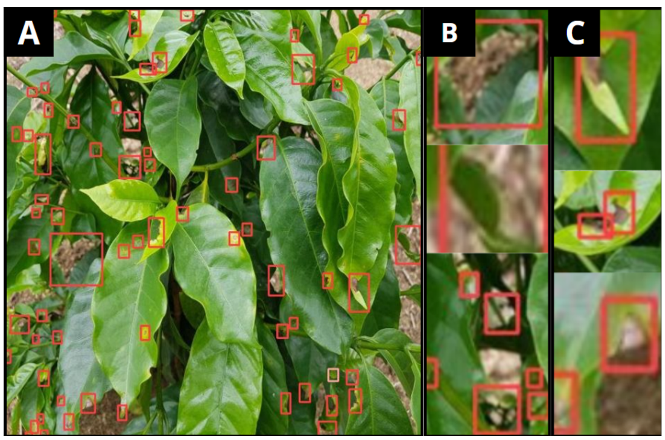

Fig. 4. Example of RT-DETR model output. (A) Detecting spots on coffee leaves in the field. (B) Examples of incorrect detection performed by the model. (C) Examples of correct detection performed by the model.

The YoloV10-M (see Figure 5 (A)) and YoloV10-X (see Figure 6 (A)) offer a good balance between accuracy and latency, achieving 0.37 and 0.36 mAP@50, respectively, with significantly lower latencies (41.2 ms and 42.5 ms) compared to RT-DETR. These models demonstrate promising performance for applications that require speed without significantly compromising accuracy. However, one drawback to consider is the relatively longer training time compared to lighter models such as YoloV10-Nano, MobileNet SSD, and EfficientDet-SSD. Additionally, Figures 5 (B) and 6 (B) show some examples of leaf spots that were not detected by these networks. On the other hand, Figures 5 (C) and 6 (C), which present zoomed-in detections, reveal that YoloV10-M handles small spots better, while YoloV10-X more easily detects larger spots.

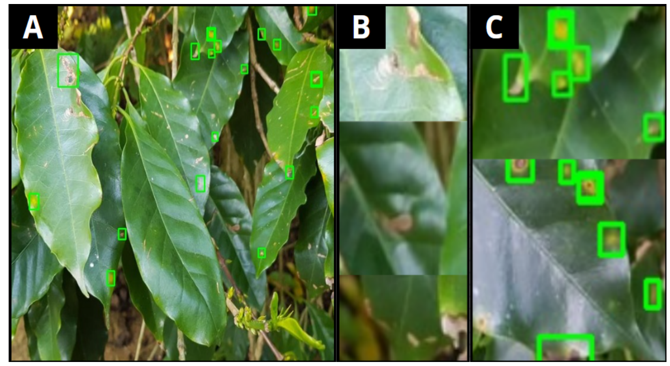

Fig. 5. Example of YoloV10-M model output . (A) Detecting spots on coffee leaves in the field. (B) Examples of spots that were not detected by the model. (C) Examples of correct detection performed by the model.

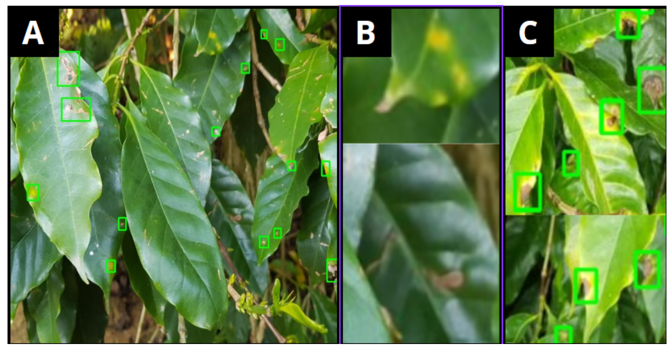

Fig. 6. Example of YoloV10-X model output (A) Detecting spots on coffee leaves in the field. (B) Examples of spots that were not detected by the model. (C) Examples of correct detection performed by the model.

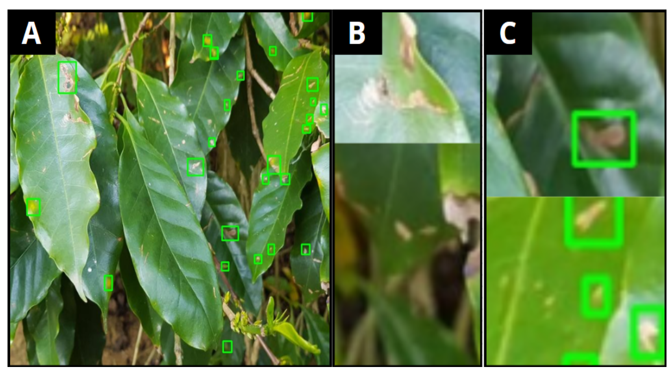

Fig. 7. Example of YoloV10-Nano model output (A) Detecting spots on coffee leaves in the field. (B) Examples of spots that were not detected by the model. (C) Examples of correct detection performed by the model.

The YoloV10-Nano (see Figure 7 (A)) stands out for its low latency (8 ms) and short training time (40 min), despite a slight drop in accuracy (0.332 mAP@50). In Figure 7 (B), it is observed that while the network does not confuse the background and detects stains of different sizes, some relevant ones are missed. In contrast, Figure 7 (C) shows correct detections made by the network, making YoloV10-Nano a viable option for hardware-constrained devices where inference speed is more critical than absolute accuracy.

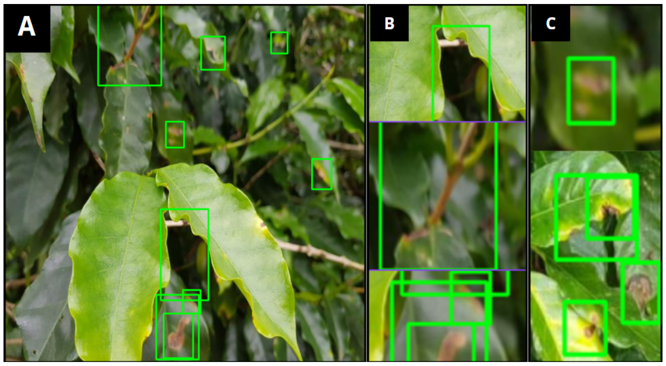

Fig. 8. Example of EfficientDet-SSD model output.(A) Detecting spots on coffee leaves in the field. (B) Examples of incorrect detection performed by the model. (C) Examples of correct detection performed by the model.

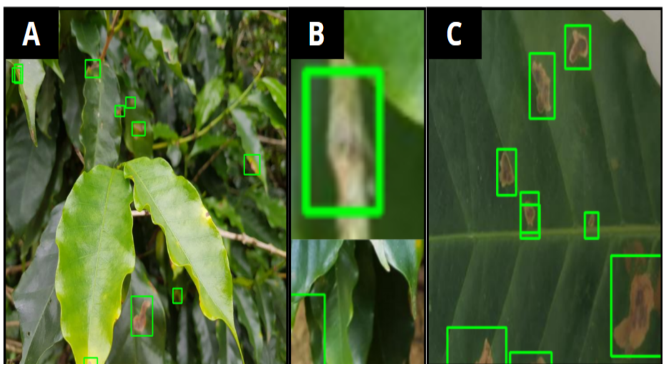

Fig. 9. Example of MobileNet-SSD model output.(A) Detecting spots on coffee leaves in the field. (B) Examples of incorrect detection performed by the model. (C) Examples of correct detection performed by the model.

The MobileNet-SSD (see Figure 9 (A)) and EfficientDet-SSD (see Figure 8 (A)) models achieved the lowest precision, with 0.26 and 0.23 mAP@50, respectively. Additionally, the EfficientDet-SSD exhibited the highest latency (275.08 ms), which is due to its generally slower inference time compared to other models [28], making it unsuitable for real-time applications. Both models face the same issue as RetinaNet and RT-DETR, confusing the background and branches with leaf spots (Figures 8 (B) and 9 (B)). EfficientDet-SSD is the most affected by this problem, as it tends to place multiple bounding boxes on the same spot, even when making good detections (Figure 8 (C)). On the other hand, despite its lower performance compared to other models, MobileNet-SSD still achieves satisfactory detections, especially in images captured in controlled environments (Figure 9 (C)).

After selecting the YoloV10-Nano model as the most suitable due to its balanced latency, mAP@50, and training time, an experimental code was developed for its execution on a mobile device. The model was embedded using Android Studio, an integrated development environment (IDE) designed for Android application development, allowing for its integration into the mobile environment. Although still in its early stages, preliminary tests indicate promising performance, highlighting its potential as a viable solution. Figure 10 provides details of its execution on the smartphone.

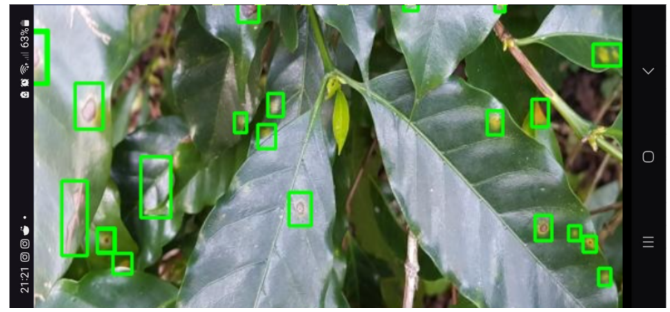

Fig. 10. Real-time detection performed by the YOLOv10-Nano model embedded in a mobile phone.

To understand how the YOLO-Nano model performed the detection of the spots, the Eigen-CAM technique was applied to different layers of the network. This technique generates activation maps for each layer, indicating the regions of the image that contributed most to the prediction. These maps allow the interpretation of the model's attention over the background, the lesions, and the contours. For this analysis, two images were used, and the results are shown in Figure 11.

In Figure 12, the map was extracted from layer 1. It can be interpreted that this layer is distinguishing the background from the spots, as the white background generated an intense red hue. The regions of the leaf without spots exhibited greenish tones, while the areas with spots did not appear to be relevant at this stage of the network. This indicates that the information extracted by layer 1 is more related to the separation between the background and the spots.

In layer 3 (Figure 13), it is noticeable that the network used this stage to distinguish the leaf from the background, delimiting its area more precisely. The model's attention began to focus on the leaf's edges, reducing the background's influence and preparing the network to identify more specific features in the subsequent layers.

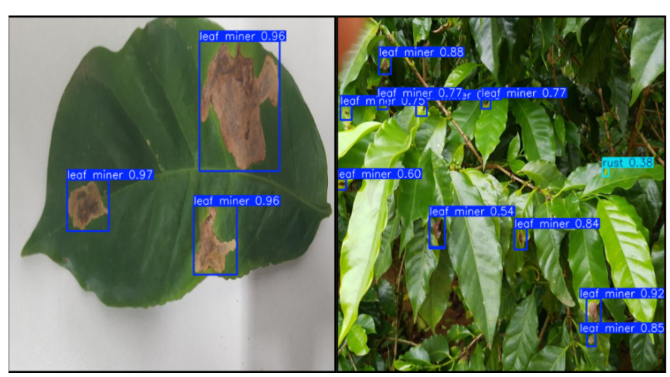

Fig. 11. Detection Results Using YOLOv10-Nano

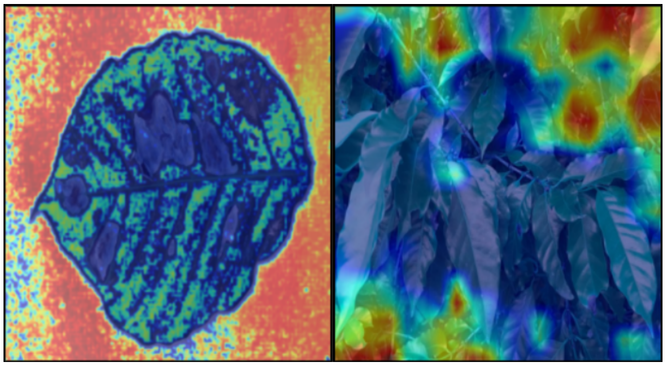

Fig. 12. Application of Eigen-CAM on the first layer of YOLOV10-Nano

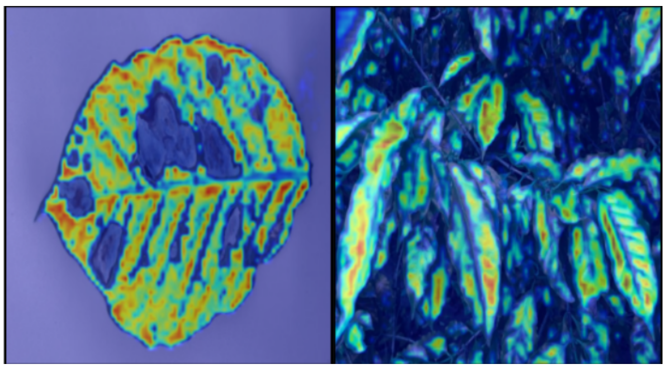

Fig. 13. Application of Eigen-CAM on the third ayer of YOLOV10-Nano

As the network becomes deeper, it begins to direct its attention to the structure of the spots. An example of this can be observed in Fig. 14, where the map was generated from the tenth layer. At this stage, it is possible to notice that the regions with spots begin to stand out with greater intensity, indicating that the network is focusing on relevant features for detecting the lesions on the leaves.

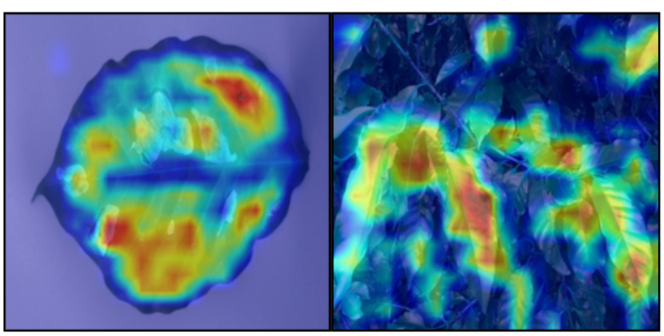

Fig. 14. Application of Eigen-CAM on the tenth layer of YOLOV10-Nano

In the twenty-first layer, represented in Fig. 15, it is noticeable that the network has already identified the regions affected by diseases with a certain level of accuracy. This can be observed by the reddish coloration over the spots, indicating that these areas contributed more significantly to the model's prediction.

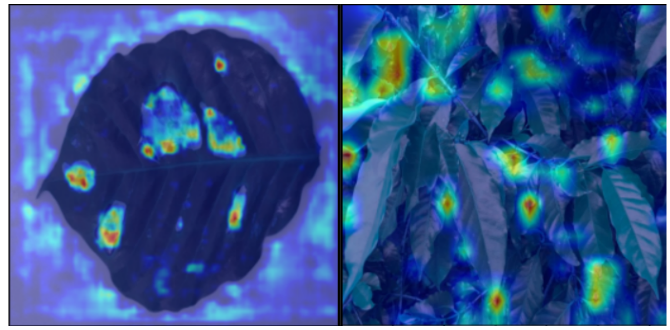

Fig. 15. Application of Eigen-CAM on the twenty-first layer of YOLOv10-Nano

Another relevant point to be observed is the influence that the number of parameters exerts on the network's performance. When comparing the maps extracted from the same layers (Fig. 16, images A, B, and C), it is possible to notice differences in the definition of the detected regions. YOLOV10-Nano, due to its more compact architecture, presents less defined maps in detection. On the other hand, YOLOV10-M, with greater capacity, shows an improvement in the delineation of these regions. Finally, the YOLOV10-X architecture, with an even higher number of parameters, presents the most defined maps, highlighting the affected areas more clearly. Even with a subtle difference in mAP values among the networks, this higher definition in the regions may influence the detection of more subtle and smaller lesions.

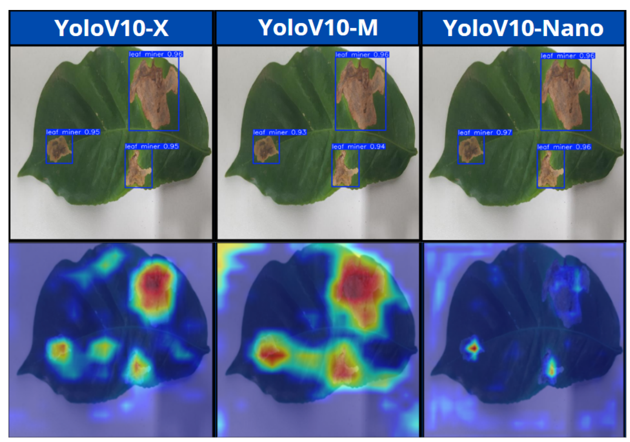

Fig. 16. Comparison of YOLOv10-X, YOLOv10-M, and YOLOv10-Nano applying Eigen-CAM on the twenty-second layer

VI. CONCLUSION

The automated detection of spots on coffee leaves, which may indicate diseases and pests, is essential given the economic relevance of coffee cultivation and the high global consumption of the beverage. This study aimed to evaluate seven DL models to identify the most suitable for the task of detecting such anomalies.

Models such as RetinaNet and RT-DETR presented high accuracy but also high latency, making them unfeasible for real-time applications. EfficientDet-SSD, in turn, showed low accuracy and also high latency. YOLOv10-Nano stood out by offering the best balance between performance and computational efficiency, with a mAP@50 of 0.332, latency of only 8 ms, and a training time of 40 minutes, making it ideal for embedded device applications.

To enhance the interpretability of YOLOv10-Nano, the Eigen-CAM technique was applied, which allows visualization of the regions in the images that most influenced the predictions. The analysis highlighted the model's focus on areas such as background, contours, and leaf spots, providing a qualitative evaluation of its decision-making process. This is essential in practical applications, as it contributes to model trust.

As future perspectives, the expansion of the dataset is highlighted, incorporating different coffee varieties, environmental conditions, and lesion stages, which may increase the model's robustness. Furthermore, it is proposed to go beyond simple detection by including the classification of associated diseases. Although the current version of YOLOV10-Nano has already been deployed on a mobile device, more in-depth tests on real hardware, such as edge devices, are necessary to validate its feasibility in the field.

ACKNOWLEDGMENTS

This research was developed as part of the project Classificação de imagens e dados via redes neurais profundas para múltiplos domínios (Image and Data Classification via Deep Neural Networks for Multiple Domains — IDeepS). The IDeepS project (available online: https://github.com/vsantjr/IDeepS, accessed on 29 July 2025) is supported by the Laboratório Nacional de Computação Científica (LNCC, MCTI, Brazil), through computational resources provided by the SDumont supercomputer. Additional support was provided by the Brazilian National Council for Scientific and Technological Development (CNPq), under process number 130346/2024-1. This study was also partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES, Brazil) — Finance Code 001.

REFERENCES

[1] Antônio Nicikava and Edemar Junior. História e consumo do café no brasil e no mundo. *Revista Interface Tecnológica*, 19:713–722, 12 2022.

[2] ConsórcioPesquisaCafé. Cafés do brasil no ano-cafeeiro 2024: faturamento bruto atinge 79,59 bilhões. http://www.consorciopesquisacafe.com.br/index.php/imprensa/noticias/1264-2025-01-15-17-58-14, 2025. Acesso em: 19 fev. 2025.

[3] PerfectDailyGrind. Raio-x do café no brasil: produção, indústria, consumo e informações relevantes sobre o mercado do país. https://perfectdailygrind.com/pt/2021/05/24/raio-x-do-cafe-no-brasil-producao-industria-consumo-informacoes-relevantes-sobre-o-mercado-do-pais/, 2021. Acesso em: 19 fev. 2025.

[4] Ajjamada C Kushalappa and Albertus B Eskes. Advances in coffee rust research. *Annual Review of Phytopathology*, 27(1):503–531, 1989.

[5] André GC Souza, Luiz A Maffia, and Eduardo SG Mizubuti. Cultural and aggressiveness variability of cercospora coffeicola. *Journal of Phytopathology*, 160(10):540–546, 2012.

[6] José Nilton Medeiros Costa, César Augusto Domingues Teixeira, José Roberto Vieira Júnior, Rodrigo Barros Rocha, and Cléberson de Freitas Fernandes. Informações para facilitar a identificação das diferentes fases do bicho-mineiro (leucoptera coffeella) em campo. https://www.infoteca.cnptia.embrapa.br/infoteca/bitstream/doc/1010232/1/cot384bichomineiro.pdf, 2012. Acesso em: 20 fev. 2025.

[7] Jorge Esquivel, Fabio Sanchez, and Luis Barboza. Infection model for analyzing biological control of coffee rust using bacterial anti-fungal compounds. *Mathematical Biosciences*, 307:13–24, 01 2018.

[8] José Aires Ventura, Hélcio Costa, and Inorbert de Melo Lima. Chapter 18 - conilon coffee diseases management. Scientific Figure on ResearchGate, 2019. Acesso em: 8 fev. 2025.

[9] Alvaro Leandro Cavalcante Carneiro, Lucas De Brito Silva, and Marisa Silveira Almeida Renaud Faulin. Artificial intelligence for detection and quantification of rust and leaf miner in coffee crop. Scientific Figure on ResearchGate, 2021. Acesso em: 8 fev. 2025.

[10] Richard Paredes-Espinosa, Joab Ricaldi, Víctor Piñan, Ana Montañez, and Ronald Rojo. *Manejo Integrado de Plagas en el cultivo de café*. 12 2022.

[11] S Santhosh Kumar, BK Raghavendra, S Ashoka, and Siddaraju. A study on core challenges in coffee plant leave disease segmentation and identification on various factors. In *ICDSMLA 2021: Proceedings of the 3rd International Conference on Data Science, Machine Learning and Applications*, pages 433–446. Springer, 2023.

[12] José G.M. Esgarioa, Renato A. Krohlinga, and José A. Venturad. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Computers and Electronics in Agriculture*, 169:105162, 2020.

[13] Thiago Vieira Machado. Aplicação de técnicas de visão computacional para identificação de ferrugem em folhas de café. https://repositorio.ufu.br/bitstream/123456789/38294/1/AplicacaoTecnicasVisao.pdf, 2023. Acesso em: 20 fev. 2025.

[14] Raka Thoriq Araaf, Arkar Minn, and Tofael Ahamed. Coffee leaf rust disease detection and implementation of an edge device for pruning infected leaves via deep learning algorithms. *Sensors*, 24(8018), 2024.

[15] Alvaro Leandro Cavalcante Carneiro, Lucas de Brito Silva, and Marisa Silveira Almeida Renaud Faulin. Artificial intelligence for detection and quantification of rust and leaf miner in coffee crop. *arXiv preprint arXiv:2103.11241*, 2021.

[16] Renato Krohling, João Eduardo Tozzi de Souza, and Lucas Miguel Tassis. Dataset sobre pragas e doenças do cafeeiro. https://data.mendeley.com/datasets/pmkbyjpf6k/1, 2021. Acesso em: 25 fev. 2025.

[17] Yuanyuan Wang, Chao Wang, Hong Zhang, Yingbo Dong, and Sisi Wei. Automatic ship detection based on retinanet using multi-resolution gaofen-3 imagery. *Remote Sensing*, 11(5):531, 2019.

[18] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16965–16974, 2024.

[19] Chien-Yao Wang, Hong-Yuan Mark Liao, et al. Yolov1 to yolov10: The fastest and most accurate real-time object detection systems. *APSIPA Transactions on Signal and Information Processing*, 13(1), 2024.

[20] Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, et al. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011, 2024.

[21] Ayesha Younis, Li Shixin, Shelembi Jn, and Zhang Hai. Real-time object detection using pre-trained deep learning models mobilenet-ssd. In *Proceedings of 2020 6th International Conference on Computing and Data Engineering*, pages 44–48, 2020.

[22] The Deep Hub. Efficientdet: An overview. https://medium.com/thedeephub/efficientdet-an-overview-c51b65ed7fcb, 2024. Acesso em: 22 fev. 2025.

[23] DataCamp. Adamw optimizer in pytorch, 2024. Accessed: 2025-03-29.

[24] D. Wilson and Tony Martinez. The need for small learning rates on large problems. volume 1, pages 115 – 119 vol.1, 02 2001.

[25] Plamen P Angelov, Eduardo A Soares, Richard Jiang, Nicholas I Arnold, and Peter M Atkinson. Explainable artificial intelligence: an analytical review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 11(5):e1424, 2021.

[26] Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review*, pages 1–66, 2022.

[27] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020.

[28] Ultralytics. Yolov10 vs. efficientdet: Object detection model comparison, 2024. Acesso em: 25 fev. 2025.