IX International Congress
of Management and Technology
COINTER PDVGT 2025

CONTAGEM AUTOMATIZADA DE OVOS DE MOSQUITOS AEDES EM OVITRAMPAS EM CONDIÇÕES REAIS USANDO VISÃO COMPUTACIONAL BASEADA EM APRENDIZADO DE MÁQUINA

CONTEO AUTOMATIZADO DE HUEVOS DE MOSQUITOS AEDES EN OVITRAMPAS EN CONDICIONES REALES USANDO VISIÓN POR COMPUTADORA BASADA EN APRENDIZAJE AUTOMÁTICO

AUTOMATED COUNTING OF AEDES MOSQUITO EGGS IN OVITRAPS UNDER REAL CONDITIONS USING MACHINE LEARNING-BASED COMPUTER VISION

Presentation: Oral Communication

Albert Alvin B. S.[1]; João Pedro P. B.[2]; Meuse Nogueira de O. J.[3]

DOI: https://doi.org/10.31692/2596-0857.IXCOINTERPDVGT.0175

RESUMO

Os mosquitos Aedes aegypti e Aedes albopictus são espécies vetoras de diversos arbovírus, como febre amarela, dengue, Zika e chikungunya. A contagem manual de ovos desses vetores é uma tarefa exaustiva e desafiadora devido à sua natureza repetitiva, demanda de tempo e necessidade de equipamentos especializados. A metodologia tradicional de coleta e contagem, frequentemente baseada em ovitrampas, embora eficaz para monitorar a presença do vetor, apresenta limitações na quantificação em larga escala. Com o aumento de casos de dengue na Região das Américas em 2024, a necessidade de fortalecer a vigilância epidemiológica torna-se urgente, sendo essencial para compreender a dinâmica populacional dos vetores, onde a contagem de ovos é uma etapa fundamental. Avanços recentes em visão computacional e aprendizado de máquina abriram novas perspectivas para automatizar tarefas complexas antes realizadas manualmente. No contexto da saúde pública, a integração dessas tecnologias promete agilizar a coleta de dados e aumentar a precisão da vigilância de vetores. A abordagem apresentada neste trabalho emprega modelos de aprendizado profundo de última geração, como YOLOv11 e Faster R-CNN, para superar os desafios inerentes à alta variabilidade na qualidade das imagens, ruídos ambientais e diferenças morfológicas sutis entre os ovos de mosquitos. O objetivo central é desenvolver um sistema automatizado para contagem precisa e eficiente de ovos, visando substituir métodos manuais. As bases teóricas fundamentam-se na entomologia aplicada à saúde pública e nos princípios de inteligência artificial para processamento de imagens. A metodologia envolveu a coleta de imagens de ovitrampas, pré-processamento de dados, treinamento dos modelos com conjuntos de dados anotados e validação comparativa. Os resultados demonstraram alta acurácia na detecção e contagem de ovos, com redução significativa do tempo de processamento em comparação com métodos tradicionais. Conclui-se que a automação baseada em aprendizado profundo pode revolucionar a vigilância de vetores, oferecendo escalabilidade e confiabilidade para programas de controle de arboviroses em cenários epidemiológicos críticos.

Palavras-Chave: *Aedes aegypti*, contagem de ovos, visão computacional, aprendizado profundo,

[1] Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal de Pernambuco, aabs2@discente.ifpe.edu.br
[2] Tecnologia em Análise e Desenvolvimento de Sistemas, Instituto Federal de Pernambuco, jppb@discente.ifpe.edu.br
[3] Doutor em Ciência da Computação, Instituto Federal de Pernambuco, meusejunior@recife.ifpe.edu.br

vigilância epidemiológica.

RESUMEN
Los mosquitos Aedes aegypti y Aedes albopictus son especies vectoras de diversos arbovirus, como fiebre amarilla, dengue, Zika y chikungunya. El conteo manual de huevos de estos vectores es una tarea exhaustiva y desafiante debido a su naturaleza repetitiva, demanda de tiempo y necesidad de equipos especializados. La metodología tradicional de recolección y conteo, frecuentemente basada en ovitrampas, aunque efectiva para monitorear la presencia del vector, presenta limitaciones en la cuantificación a gran escala. Con el aumento de casos de dengue en la Región de las Américas en 2024, la necesidad de fortalecer la vigilancia epidemiológica se vuelve urgente, siendo esencial para comprender la dinámica poblacional de los vectores, donde el conteo de huevos es una etapa fundamental. Avances recientes en visión por computadora y aprendizaje automático han abierto nuevas perspectivas para automatizar tareas complejas antes realizadas manualmente. En el contexto de la salud pública, la integración de estas tecnologías promete agilizar la recolección de datos y aumentar la precisión de la vigilancia vectorial. El enfoque presentado en este trabajo emplea modelos de aprendizaje profundo de última generación, como YOLOv11 y Faster R-CNN, para superar los desafíos inherentes a la alta variabilidad en la calidad de las imágenes, ruidos ambientales y diferencias morfológicas sutiles entre los huevos de mosquitos. El objetivo central es desarrollar un sistema automatizado para conteo preciso y eficiente de huevos, buscando reemplazar métodos manuales. Las bases teóricas se fundamentan en la entomología aplicada a la salud pública y los principios de inteligencia artificial para procesamiento de imágenes. La metodología involucró la recolección de imágenes de ovitrampas, preprocesamiento de datos, entrenamiento de modelos con conjuntos de datos anotados y validación comparativa. Los resultados demostraron alta precisión en la detección y conteo de huevos, con reducción significativa del tiempo de procesamiento en comparación con métodos tradicionales. Se concluye que la automatización basada en aprendizaje profundo puede revolucionar la vigilancia vectorial, ofreciendo escalabilidad y confiabilidad para programas de control de arbovirosis en escenarios epidemiológicos críticos.

Palabras Clave: *Aedes aegypti*, conteo de huevos, visión por computadora, aprendizaje profundo, vigilancia epidemiológica.

ABSTRACT
The mosquitoes Aedes aegypti and Aedes albopictus are vector species for several arboviruses, such as yellow fever, dengue, Zika, and chikungunya. Manual egg counting of these vectors is an exhaustive and challenging task due to its repetitive nature, time consumption, and need for specialized equipment. Traditional collection and counting methodologies, often based on ovitraps, though effective for monitoring vector presence, have limitations in large-scale quantification. With the rising number of dengue cases in the Region of the Americas in 2024, the need to strengthen epidemiological surveillance is urgent, as it is essential for understanding vector population dynamics, where egg counting is a fundamental step. Recent advances in computer vision and machine learning have opened new avenues for automating complex manual tasks. In public health, integrating these technologies promises to streamline data collection and enhance vector surveillance precision. The approach presented in this work leverages state-of-the-art deep learning models, such as YOLOv11 and Faster R-CNN, to overcome inherent challenges like high image-quality variability, environmental noise, and subtle morphological differences between mosquito eggs. The core objective is to develop an automated system for accurate and efficient egg counting to replace manual methods. Theoretical foundations are grounded in public health entomology and artificial intelligence principles for image processing. The methodology involved collecting ovitrap images, data preprocessing, training models with annotated datasets, and comparative validation. Results demonstrated high accuracy in egg detection and counting, with significantly reduced processing time compared to traditional methods. It is concluded that deep learning-based automation can revolutionize vector surveillance, offering scalability and reliability for arbovirus control programs in critical epidemiological scenarios.

Keywords: *Aedes aegypti*, egg counting, computer vision, deep learning, epidemiological surveillance.

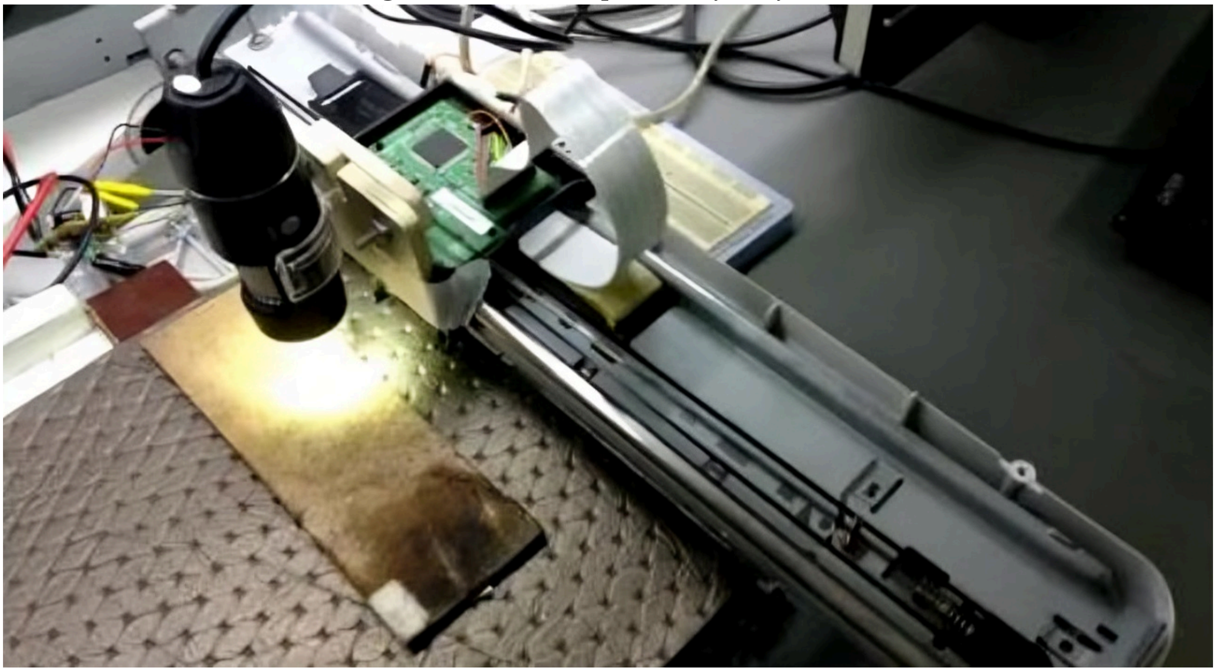
INTRODUCTION

Mosquito-borne arboviruses—most notably dengue, Zika and chikungunya—continue to exert a heavy burden on public health systems, especially in tropical and subtropical regions where *Aedes aegypti* and *Aedes albopictus* thrive. Surveillance based on ovitraps is a cornerstone of entomological monitoring because the number of eggs collected provides an early, low-cost indicator of vector presence and reproduction potential. However, in practice the step that converts ovitrap palettes into usable numeric indicators—the manual visual counting of eggs—remains laborious, slow and susceptible to inter-operator variability, reducing the timeliness and comparability of surveillance outputs during routine monitoring and outbreaks. These operational constraints motivate the search for scalable, automated tools that preserve epidemiological value while reducing human effort.

This study responds to that need by developing and evaluating a practical, end-to-end automated pipeline for detecting and counting Aedes eggs on ovitrap palettes using modern computer-vision methods. The research rests on a real, curated image corpus provided by Fiocruz-PE and captured with an accessible hardware stack: an OI1004 USB digital microscope (up to 1600×, 8 white LEDs) controlled via HiView and mounted on a low-cost conveyor system repurposed from inkjet-printer servomechanisms to enable systematic, repeatable captures. The original acquisition yielded 1,471 images (640×480 px); after manual curation to remove duplicates and invalid frames the working dataset totals 1,195 images, which were split 80/20 for training and validation. Annotation was carried out manually in CVAT, producing bounding-box labels that were converted for use with the chosen training frameworks. These concrete dataset and capture details both ground the experiments in real laboratory constraints and expose the kinds of noise and variability (dirt accumulation, specular reflections on eggs, focal blur from palette deformation and occasional non-target eggs) that must be handled for any practical, operational system.

From a technical perspective the problem presents two overlapping challenges. First, eggs are very small (on the order of 0.4–1.0 mm) and often densely clustered on the palette, which strains object detectors' small-object localization capabilities and increases the risk of merged or missed detections. Second, the imaging artifacts present in the dataset—variable illumination, reflections, substantial dirt, deformation blur and confounding objects—break assumptions common to classical, thresholding-based pipelines and require methods with strong invariance and representational power. These conditions motivated the decision to favor convolutional neural network detectors and transfer-learning strategies over classical segmentation, and to emphasize augmentation, preprocessing and careful error analysis in the experimental design.

Figure 01: Microscope Conveyor System.



Source: Own (2025).

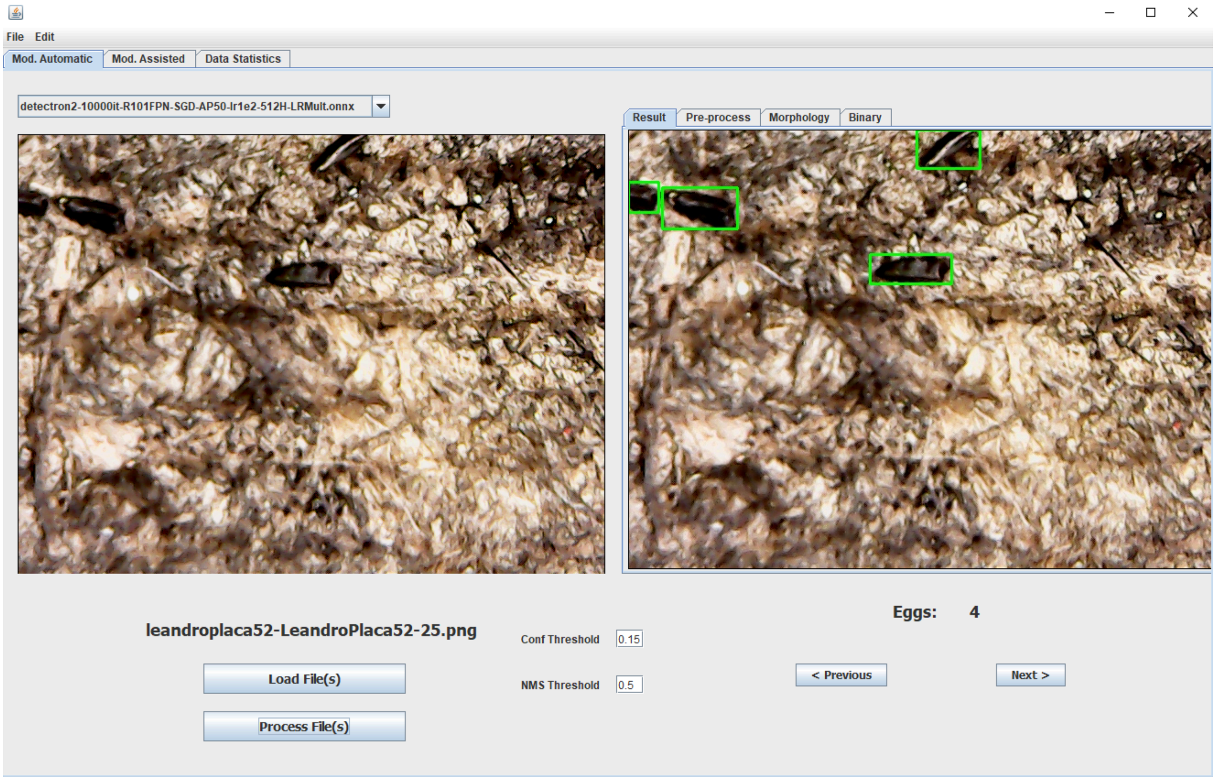
The central hypothesis guiding this work is that contemporary CNN-based object detectors—when fine-tuned on a domain-specific, curated dataset and supported by targeted preprocessing and augmentation—can deliver reliable, reproducible egg counts that substantially cut manual workload and maintain or improve the accuracy of entomological indicators. To test this hypothesis we compare modern single-stage detectors (YOLOv10 and YOLOv11) and a two-stage detector (Faster R-CNN with optional FPN variants), applying transfer learning from standard backbones, carefully tuning hyperparameters and recording standardized detection metrics (Precision, Recall, mean Average Precision at IoU thresholds such as mAP@50 and the stricter mAP@50–95, plus a composite "fitness" measure used during hyperparameter searches). Training experiments were executed on cloud platforms (Lightning AI for YOLO variants and Kaggle for Detectron2/Faster R-CNN experiments) and all artifacts were versioned for reproducibility.

Specific objectives flow directly from this hypothesis and the operational constraints of laboratory deployment: (1) prepare and curate a representative, well-documented ovitrap image dataset and annotation set; (2) implement preprocessing and a suite of augmentations that emulate observed noise patterns (brightness/contrast jitter, small rotations, Gaussian blur, random erasing to simulate dirt/occlusion and mild geometric distortions); (3) fine-tune and

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

compare YOLOv10, YOLOv11 and Faster R-CNN under a shared evaluation protocol; (4) analyze failure modes in detail to identify systematic error sources and data-collection remedies; and (5) demonstrate a practical integration pathway by exporting the best model to ONNX and embedding inference into an existing Java Swing application (CataOvoMaven) to enable batch processing and an operator-friendly GUI. These objectives are operationally motivated—each step maps to a concrete barrier that currently prevents routine adoption of automated counting in many laboratories.

Figure 02: CataOvoMaven's main screen.



Source: Own (2025).

Justification for the present project combines epidemiological urgency and a clear technological gap. Entomological indicators derived from ovitraps are central to many vector-control strategies, yet manual counting limits sample throughput and introduces variability that can obscure real trends at early outbreak stages. Moreover, earlier automated approaches based on thresholding and classical image processing, while useful in idealized conditions, fail under the palette-level artifacts documented in our corpus; recent CNN-based solutions (for example, iCount and Egg-countAI) demonstrate the feasibility of deep-learning approaches but are typically validated on cleaner or differently-acquired imagery, leaving a gap in understanding how detectors behave on noisy, operational datasets like those produced by Fiocruz-PE. By working with a realistic dataset, comparing multiple state-of-the-art detectors under the same protocol and showing a path to software integration, this work seeks to bridge that gap and make adoption practical for public-health laboratories.

Methodologically, the study combines careful dataset curation and annotation guidelines with a systematic training and evaluation pipeline. The dataset preparation stage included Python scripts to shuffle and split images reproducibly and to remove duplicates or frames that did not match ovitrap standards; annotation guidelines standardized how to handle partial occlusion, minimal box sizes and ambiguous cases so that label noise is minimized. During model development we emphasized controlled ablation studies—testing augmentation sets, optimizer choices (AdamW vs. SGD schedules), anchor and NMS parameter sweeps, and backbone options (ResNet variants with or without FPN)—to identify robust settings rather than one-off wins. Evaluation paired quantitative reporting (Precision/Recall, mAP metrics and inference-time measurements) with qualitative failure analysis (annotated examples of reflections yielding false positives, clustered eggs producing merged boxes, and major missed detections caused by severe blur) so that experimental conclusions translate into actionable recommendations for data collection and deployment.

The contribution of this article is therefore both empirical and practical. Empirically, it documents a reproducible pipeline and a comparative evaluation that clarifies the strengths and weaknesses of contemporary detectors when applied to realistic ovitrap imagery; practically, it demonstrates a concrete route to adoption—exporting a trained model to ONNX and integrating it with the CataOvoMaven Java interface—so that laboratories can perform batch inference, visualize bounding boxes and collect per-palette counts with minimal operational change. The article also explicitly records limitations—dataset size and seasonal breadth, the single-class (egg) labeling choice that conflates A. aegypti and A. albopictus eggs, and the need for field trials to validate operational usefulness—and it proposes prioritized next steps (dataset expansion across seasons and field sites, multi-class labeling to separate confounders, model pruning/quantization for edge deployment and formal operational trials). These candid limitations are intended to make the reported results useful and actionable rather than prematurely definitive.

THEORETICAL FOUNDATION

The eggs of Aedes aegypti and Aedes albopictus share dominant visual traits — oval/elliptical shape (≈0.5–1.0 mm) and a reticulated chorion — while showing subtle ornamentation differences. These morphological features (shape, texture, contrast, chorion pattern) define the target object class and justify modeling eggs as a single unified class for detection. Understanding which morphological cues are invariant versus species-specific informs feature extraction, annotation strategy, and error analysis.

Classical computer vision (filters, geometric transforms, thresholding/Otsu) provides interpretable preprocessing and segmentation tools but is sensitive to noise, illumination and artifacts commonly present in field images. Modern computer vision (deep learning/CNNs) learns hierarchical features directly from data, offering robustness to scale, rotation and occlusion. The theoretical choice between handcrafted methods and learned representations shapes preprocessing, model selection and expectations about robustness in real-world conditions.

Supervised learning is the chosen paradigm: labeled images and bounding boxes teach models to map pixels to object locations. Deep learning, particularly convolutional neural networks, supplies the representational power to capture local textures and global shapes essential for egg detection. Transfer learning and fine-tuning of pre-trained networks accelerate convergence and improve generalization when labeled data are limited.

Single-stage detectors (YOLO family) prioritize speed by predicting boxes and classes in one pass, suitable for real-time needs but potentially weaker on very small or crowded objects. Two-stage detectors (Faster R-CNN + RPN, FPN) emphasize localization precision and scale robustness at higher computational cost. The theoretical trade-off between latency and accuracy guides model selection according to deployment requirements and the expected image complexity.

Optimization choices (AdamW vs SGD) affect convergence dynamics and the tendency to find flat vs sharp minima, which in turn influence generalization. Fine-tuning strategies (freeze early layers; adapt later layers) leverage pre-learned generic features while specializing for the egg-detection domain. Standards like ONNX are adopted to enable model portability across runtimes, with the acknowledgment that conversion may incur small performance differences.

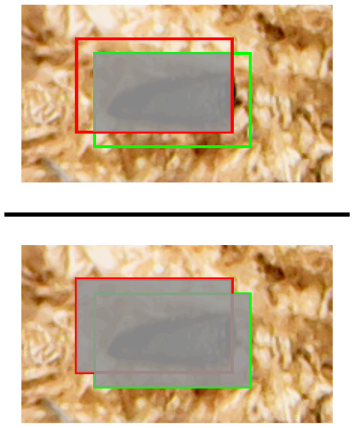
Detection performance is measured with localization-aware metrics (IoU, mAP@50, mAP@50–95), complemented by precision/recall, F1, PR curves and precision/recall vs confidence analyses. For a single-class dataset (eggs), AP/mAP remain valid but must be interpreted in that constrained context. A robust evaluation protocol (hold-out test set, appropriate IoU thresholds, PR curves) is essential to support reliable analysis and to compare models and preprocessing variants.

- **IoU (Intersection over Union):** Measures overlap between predicted and ground-truth boxes. Higher = better localization.

- **mAP@50:** Mean Average Precision at IoU ≥ 0.5; measures detection accuracy with a lenient overlap threshold.
- **mAP@50–95:** Mean AP averaged over IoU thresholds from 0.5 to 0.95; stricter, more robust accuracy measure.
- **Precision:** Fraction of predicted positives that are correct (low false positives).
- **Recall:** Fraction of true objects that are detected (low false negatives).
- **F1 score:** Harmonic mean of precision and recall, balancing both.
- **PR curve:** Precision–Recall trade-off across confidence thresholds.
- **Precision/Recall vs Confidence:** Shows how prediction certainty affects errors and misses.

Figure 03: Intersection Over Union (IoU) visual demonstration.

$$\mathbf{IoU} = \frac{\textit{Intersection area}}{\textit{Union area}} =$$



Source: Nogueira de O. et al. (2021).

Several works demonstrate practical, low-cost solutions based on thresholding, morphological operators and handcrafted feature rules. These methods (e.g., controlled light-box acquisition pipelines and mobile apps) can perform well under controlled imaging conditions because they exploit high foreground–background contrast and predictable egg appearances. However, they typically require manual parameter tuning (thresholds, size filters) and are sensitive to dirt, reflections, blur and uneven illumination — factors common in field images — which increase false positives/negatives and limit generalizability.

Existing software tools that implement classical pipelines enable fast counts in many scenarios but encounter difficulties with high egg density, overlapping objects and noisy substrates. They often demand per-image or per-batch parameter configuration by the user, which reduces throughput and reproducibility when processing large field datasets. Reports highlight trade-offs between ease-of-use/cost and applicability to uncontrolled environments.

More recent methods apply deep networks (R-CNN variants, object detectors) to egg

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

counting and achieve high accuracy in microscopic or well-controlled macro images. These approaches substantially reduce manual preprocessing and subjectivity and can reach accuracies reported above 98% in laboratory datasets. Yet, most published DL models were trained on micro- or macro-images of strips/plates captured under standard conditions; consequently they often underperform when faced with domain shifts introduced by outdoor palettes (soil, debris, varied lighting, deformation).

Comparative literature and benchmarks show the familiar pattern: single-stage detectors (YOLO family) excel at throughput and are favorable for real-time deployment, while two-stage detectors (Faster R-CNN with FPN) usually provide better localization and handling of small or crowded objects at the cost of inference speed. Many studies emphasize the importance of scale-aware backbones (FPN, dilated convolutions) and multi-scale training/augmentation to improve small-object detection — directly relevant for eggs.

The main gap identified across prior works is a mismatch between datasets used for model development and the real conditions where automated counting is desired. Field images introduce noise patterns and object arrangements (clustered eggs, substrate irregularities) not present in laboratory datasets. This raises several methodological needs: curated field-representative datasets, robust augmentation strategies, domain-adaptation or fine-tuning on field samples, and evaluation protocols that explicitly measure robustness to common perturbations (noise, occlusion, brightness changes).

METHODOLOGY

The research has an applied and qualitative nature, structured as a case study. It is situated in the field of Information Technology, specifically in the development and application of computer vision and machine learning techniques for public health. The subjects of the research are the images of mosquito eggs collected from ovitraps, provided by Fiocruz-PE, which served as the dataset for model training and evaluation. The instruments used include digital microscopes for image acquisition, the CVAT tool for image annotation, and computational frameworks such as YOLO and Faster R-CNN implemented with Ultralytics and Detectron2. The procedure involved several steps: the collection and preparation of the dataset, the labeling of images, the application of data augmentation, the training and fine-tuning of deep learning models, and finally the integration of the trained models into a Java-based application for automated counting. The evaluation was carried out using standard performance metrics such as precision, recall, mean average precision (mAP), and F1-score to validate the effectiveness of the proposed system.

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

Technological Resources

The desktop system for automatic detection and counting of Aedes eggs was developed in Java 17, selected for its stability and portability. Supporting technologies and tools included:

- Maven – project dependency management and build automation
- Java Swing – graphical user interface (GUI) development
- OpenCV – image manipulation, preprocessing, and visualization
- CVAT – collaborative labeling tool for creating ground-truth bounding boxes
- YOLOv10 and YOLOv11 – object detection architectures, accessed via Ultralytics Python SDK
- Detectron2 – framework for training Faster R-CNN models
- NetBeans IDE – main development environment for Java code
- VS Code – used for Python scripting and experimentation
- Lightning AI and Kaggle – cloud-based platforms for model training and fine-tuning
- Matplotlib – for generating evaluation charts and visualizing performance metrics
- ONNX Runtime – ensuring model portability between Python training environments and Java deployment

Dataset Acquisition and Preparation

The dataset was provided by Fiocruz Pernambuco and consisted of images of oviposition pallets obtained from ovitraps. Image capture was performed using a digital USB microscope OI1004 (maximum magnification: 1600×), mounted on a conveyor system built from recycled printer components for automated positioning. Initially, 1,471 images were collected at a resolution of 640×480 pixels. The images were curated to remove low-quality or irrelevant samples, resulting in a final dataset of 1,195 images. These were split into:

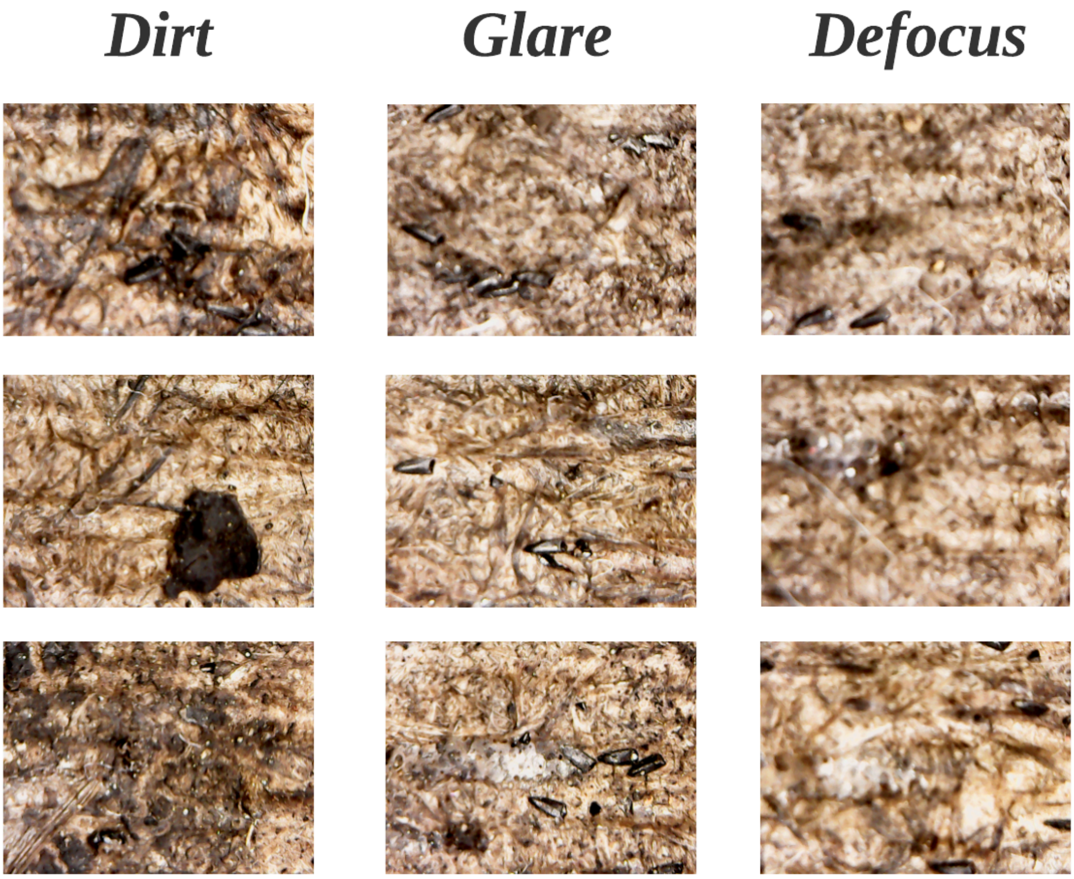
- Training set: 956 images;
- Validation set: 239 images.

The captured images contained several challenges typical of field conditions, including dirt and debris on pallets, light reflections, motion blur, overlapping eggs, and eggs from other insect species. To create reliable ground-truth annotations, the CVAT tool was used to draw bounding boxes around eggs, exporting both YOLO and COCO format labels for compatibility with the chosen architectures.

Figure 04: Examples of validation images containing visual artifacts—such as glare, dirt, and defocus—that negatively impacted model performance. These challenging conditions illustrate

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

common sources of error for algorithm variations during inference.



Source: Own (2025).

Model Training Procedures

Two model families were evaluated: YOLOv10/YOLOv11 and Faster R-CNN (via Detectron2). For the YOLO architectures, training was performed in the Lightning AI environment, which uses an NVIDIA A10G GPU with 22,723 MB of total memory and 80 multiprocessors, providing a robust environment for fine-tuning the models. On the other hand, the training of the Faster R-CNN was conducted on Kaggle, using a Tesla P100-PCIE-16GB GPU with 16,269 MB of total memory and 56 multiprocessors.

YOLO Training:

- Tested small, medium, and large variants, starting with the smallest model for initial evaluation and scaling up based on results.
- SGD optimizer: 1,000 epochs, batch size 32, momentum 0.9, weight decay 0.0005, learning rate 0.01
- AdamW optimizer: 600 epochs, same batch/momentum/weight decay, learning rate 0.002

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

- Image size fixed at 640×640 pixels for consistency with model input requirements.

Faster R-CNN Training:

- Backbones with FPN and DC5 configurations were tested.
- SGD: 10,000 iterations
- AdamW: 6,500 iterations
- Hyperparameters aligned with YOLO experiments for comparability.

Data Augmentation was applied to improve generalization and robustness:

- Geometric: random horizontal/vertical flips, scaling, and cropping
- Photometric: brightness/contrast adjustment, lighting variation
- Spatial: randomized crop regions simulating partial occlusion

Model Integration and Application Workflow

After training, the best-performing models were exported to ONNX format to ensure compatibility with the Java-based desktop application. The integration pipeline included:

- OpenCV loading the image from file or camera feed
- Conversion from BGR to RGB (matching training input format)
- Resizing to 640×640 pixels
- Pixel normalization before passing to the ONNX model
- Post-processing of model outputs to extract bounding boxes, confidence scores, and counts
- Displaying annotated images and egg counts directly in the application interface

This approach enabled the trained models to be executed efficiently within a Java environment without requiring Python dependencies during runtime.

RESULTS AND DISCUSSION

The evaluated models showed distinct performances in terms of precision, recall, and aggregated metrics such as mAP@50 and mAP@50-95, with their calculations based on the validation dataset. Table 1 presents the results for the YOLO architecture, while Table 2 displays the results for Faster R-CNN.

Table 01: Performance results of the YOLO architecture.
Precision and Recall were evaluated at IoU = 0.7 and Confidence = 0.5.

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

Model	Precision	Recall	mAP50	mAP50-95	Fitness
YOLOv10-S-SGD	0.9213	0.8367	0.8952	0.6660	0.6889
YOLOv10-S-AdamW	0.9405	0.8265	0.8980	0.6564	0.6806
YOLOv11-L-SGD	0.9477	0.9254	0.9514	0.7085	0.7328
YOLOv11-L-AdamW	0.9192	0.9286	0.9476	0.6979	0.7229
YOLOv11-M-SGD	0.9455	0.9184	0.9483	0.6992	0.7241
YOLOv11-M-AdamW	0.9354	0.9184	0.9525	0.6925	0.7185
YOLOv11-S-SGD	0.9772	0.8740	0.9407	0.6736	0.7003
YOLOv11-S-AdamW	0.9631	0.9286	0.9588	0.6968	0.7230

Source: Own (2025).

Table 02: Performance results of the Faster R-CNN architecture.
Precision and Recall were evaluated at IoU = 0.7 and Confidence = 0.5.

Model	Precision	Recall	mAP50	mAP50-95	Fitness
Faster-RCNN-R101-FPN-x3-AdamW	0.7264	0.9059	0.8125	0.5773	0.6008
Faster-RCNN-R101-FPN-x3-SGD	0.6636	0.8588	0.8416	0.5746	0.6013
Faster-RCNN-R50-FPN-x3-AdamW	0.7723	0.9176	0.8170	0.6003	0.6275
Faster-RCNN-R50-FPN-x3-SGD	0.6991	0.9294	0.8725	0.6003	0.6275
Faster-RCNN-R50-DC5-x3-AdamW	0.7499	0.9176	0.8084	0.5467	0.5728
Faster-RCNN-R50-DC5-x3-SGD	0.6944	0.8823	0.8163	0.5464	0.5734

Source: Own (2025).

The analysis of precision and recall metrics reveals that the YOLOv11 architecture models achieve the best performance. In particular, YOLOv11-S-SGD obtained the highest precision (97.72%), while YOLOv11-L-AdamW and YOLOv11-S-AdamW achieved the highest recall (92.86%). On the other hand, in the Faster R-CNN architecture, the Faster-RCNN-R50-FPN-x3-SGD model obtained the highest recall (92.94%), while Faster-RCNN-R50-FPN-x3-AdamW recorded the highest precision (77.23%). These results suggest that Faster R-CNN struggles to reach comparable performance levels, highlighting opportunities for future optimizations.

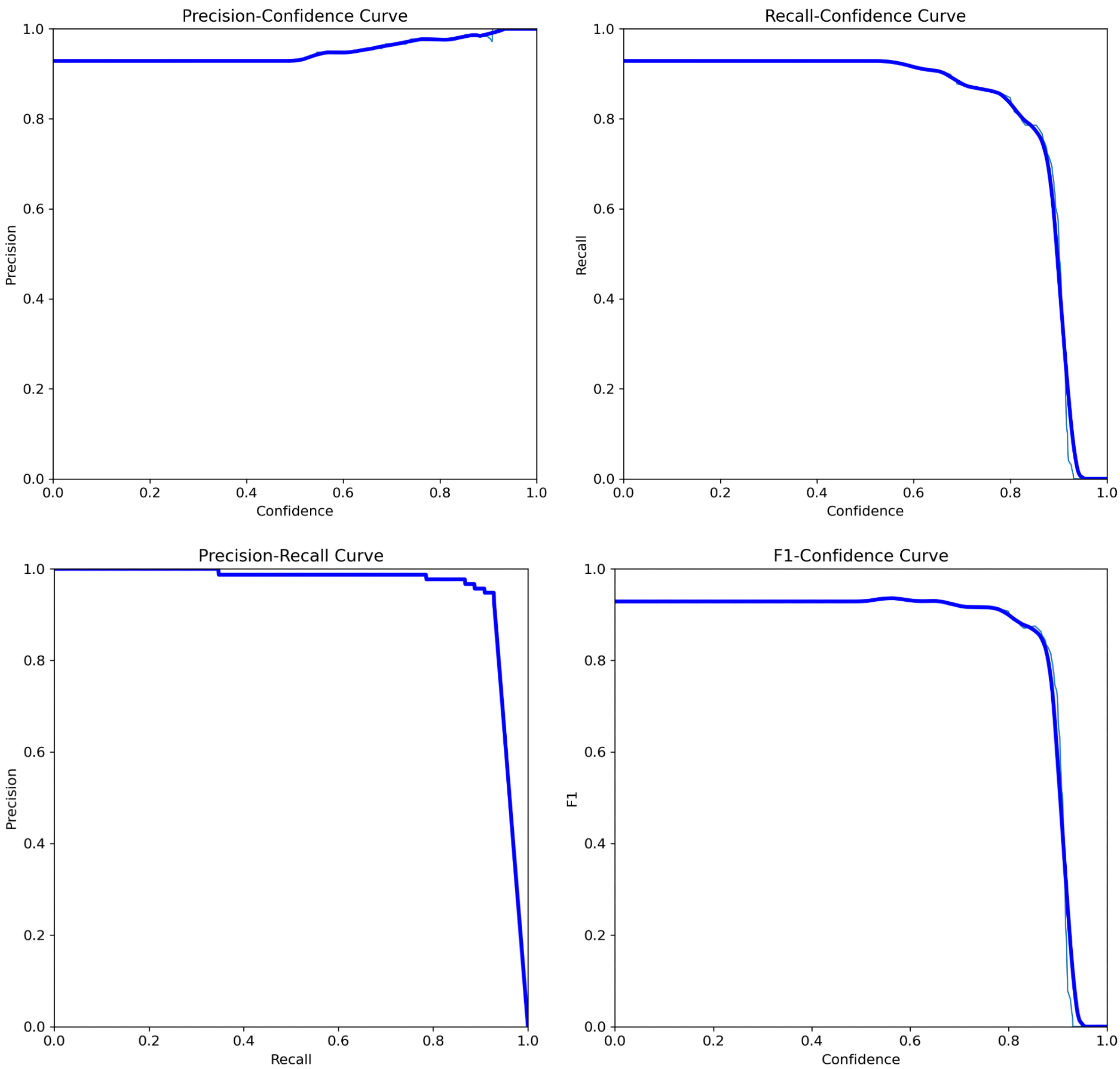
The YOLOv11-M-AdamW and YOLOv11-S-AdamW models achieved the highest mAP@50 scores—95.25% and 95.88%, respectively—demonstrating excellent object detection performance. For mAP@50-95, YOLOv11-L-SGD delivered the best result (70.85%), highlighting its superiority in object detection across multiple IoU thresholds. As for the Faster R-CNN architecture, the highest mAP@50 and mAP@50-95 scores were attained by Faster-RCNN-R50-FPN-x3-SGD, at 87.25% and 60.03%, respectively.

The fitness metric, which combines multiple performance factors, revealed that the

YOLOv11-L-SGD (73.28%) and YOLOv11-M-SGD (72.41%) models were the most balanced in terms of overall performance. Among the Faster R-CNN models, the best fitness score was observed for Faster-RCNN-R50-FPN-x3-SGD (62.75%).

Figure 04: Evaluation Metrics for YOLOv11-L-SGD on Validation Data

Precision and Recall were evaluated at IoU = 0.7 and Confidence = 0.5.



Source: Own. (2025).

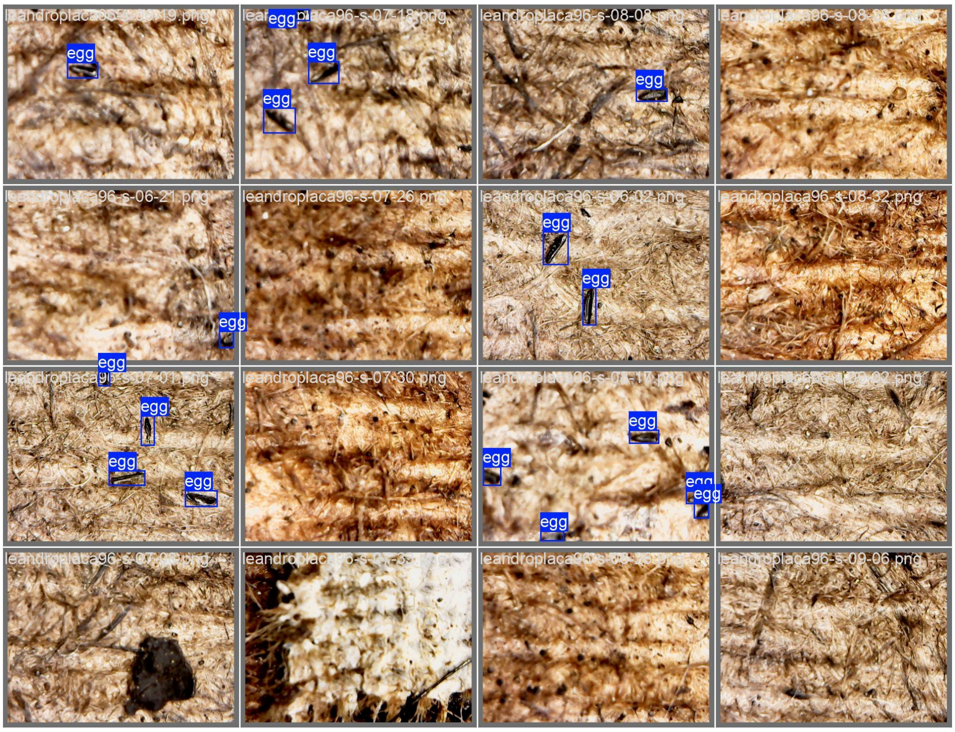
Based on the analyzed metrics, we conclude that the YOLOv11 architecture models stand out as the most balanced and efficient in terms of overall performance. In particular, YOLOv11-L-SGD and YOLOv11-L-AdamW emerge as the top choices, combining high precision, recall, and robust mAP@50 and mAP@50-95 values - demonstrating their ability

ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857

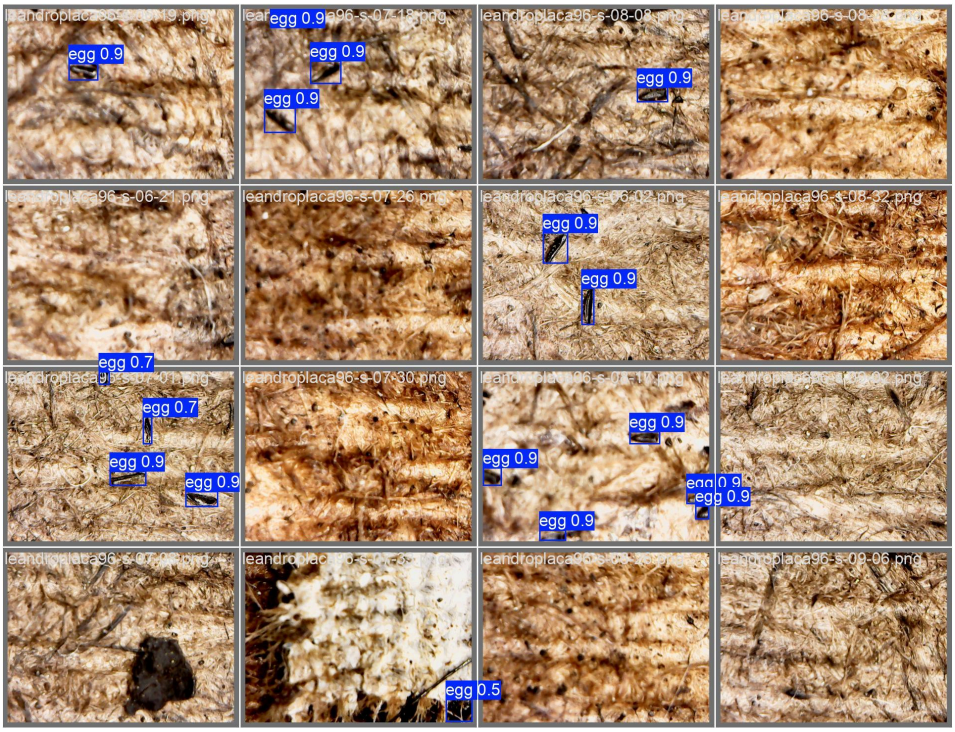
to balance multiple performance aspects while maintaining strong generalization capability across different IoU thresholds. Notably, YOLOv11-L-SGD was the only model to achieve mAP@50-95 above 0.7 (70.85%).

Conversely, while the Faster R-CNN architecture showed competitive recall results (92.94% for Faster-RCNN-R50-FPN-x3-SGD), its overall performance lagged behind YOLOv11 models, particularly in precision and mAP@50-95 metrics. This suggests Faster R-CNN still faces significant challenges in achieving an optimal precision-recall balance, highlighting clear areas for future optimization.

Figure 05: Comparison of Ground Truth and YOLOv11-L-SGD Predictions on Validation Data. Precision and Recall were evaluated at IoU = 0.7 and Confidence = 0.5.



ISSN: 2596-0857
PREFIXO DOI: 10.31692/2596-0857



Source: Own. (2025).

CONCLUSIONS

The proposed solution, based on deep learning for the automated detection and counting of Aedes eggs, addressed the research objective of improving entomological surveillance methods and demonstrated that modern neural networks can outperform both classical computer vision techniques and other deep learning approaches previously applied in this field.

The findings confirm that YOLOv11-L-SGD provided the best balance of accuracy and robustness, achieving remarkable performance even under noisy, variable, and challenging imaging conditions. This result directly supports the initial hypothesis that a carefully adapted deep learning architecture could significantly advance the automation of vector monitoring. The successful integration of the trained model into the CataOvoMaven application further validates the practical feasibility of the proposed approach, closing the gap between theoretical research and real-world application.

At the same time, the research highlighted several limitations that must be acknowledged. The dataset, while representative, was not large enough to cover the full variability of field conditions, which may affect generalization. The lack of separate labeling for non-Aedes eggs could generate false positives in more complex scenarios. Computational cost also emerged as a limiting factor, especially for large models such as YOLOv11-L, which require robust hardware for real-time inference. In addition, the reliance on free platforms for training imposed constraints on hyperparameter optimization, and the ONNX conversion process introduced minor performance losses. These challenges indicate that future studies should invest in more flexible or premium infrastructures, as well as more refined tuning, particularly for architectures like Faster R-CNN, which showed untapped potential under better conditions.

Despite these constraints, the contributions of this research are significant. It introduced an innovative, automated pipeline for mosquito egg counting that reduces analysis time, eliminates human subjectivity, and produces standardized entomological data — a valuable asset for monitoring mosquito populations and evaluating vector control strategies. By applying advanced techniques such as transfer learning, fine-tuning, and data augmentation, the work demonstrated how deep learning can be adapted to highly specific public health problems. Moreover, the replicable methodology presented here provides a model for extending convolutional neural networks to other tasks of vector detection and surveillance.

In conclusion, this study not only confirms the potential of deep learning to overcome practical challenges in entomological surveillance but also advances the broader use of artificial intelligence in public health. By offering a robust, accessible, and scalable solution, the research contributes to the fight against arboviruses such as dengue, zika, and chikungunya in endemic regions. The results obtained answer the research objectives positively, while also opening perspectives for future refinements and broader applications of AI-driven surveillance systems.

REFERENCES

ABOYOMI, D.; DANIEL, C. A comparative analysis of modern object detection algorithms: yolo vs. ssd vs. faster r-cnn. ITEJ (Information Technology Engineering Journals), 8(2):96–106, Dec. 2023. doi: 10.24235/itej.v8i2.123. URL https://www.syekhnurjati.ac.id/journal/index.php/itej/article/view/123.

ALVES, L. D. Desenvolvimento de um sistema de baixo custo para contagem automática de ovos de Aedes aegypti utilizando técnicas de processamento de imagem. 2016. Dissertação (Mestrado). Fundação Getulio Vargas, Escola de Matemática Aplicada. URL https://repositorio.fgv.br/items/f1255e89-b525-49c0-b639-35150ea2d199.

LIU, L.; WANG, A.; CHEN, H.; et al. Yolov10: real-time end-to-end object detection. arXiv preprint arXiv:2405.14458, 2024.

BRAGA, I. A.; VALLE, D. Aedes aegypti: histórico do controle no Brasil. Epidemiologia e Serviços de Saúde, 16:113–118, Jun. 2007. ISSN 1679-4974.

DONALÍSIO, M. R.; GLASSER, C. M. Vigilância entomológica e controle de vetores do dengue. Revista Brasileira de Epidemiologia, 5(3):259–279, Dec. 2002. ISSN 1415-790X. doi: 10.1590/S1415-790X2002000300005. URL https://doi.org/10.1590/S1415-790X2002000300005.

APACHE SOFTWARE FOUNDATION. NetBeans 22. 2025. URL https://netbeans.apache.org/.

GABURRO, J.; DUCHEMIN, J.-B.; PARADKAR, P. N.; NAHAVANDI, S.; BHATTI, A. Assessment of iCount software, a precise and fast egg counting tool for the mosquito vector Aedes aegypti. Parasites & Vectors, 9(1):590, Nov. 2016. ISSN 1756-3305. doi: 10.1186/s13071-016-1870-1. URL https://doi.org/10.1186/s13071-016-1870-1.

HUNTER, J. D. Matplotlib: a 2D graphics environment. 2007. URL https://matplotlib.org/.

INTEL CORPORATION. CVAT: Computer Vision Annotation Tool. 2018. URL https://cvat.ai/. Acessado em: 31 dez. 2024.

JAVED, N.; LÓPEZ-DENMAN, A. J.; PARADKAR, P. N.; BHATTI, A. Egg-countAI: a convolutional neural network-based software for counting of Aedes aegypti mosquito eggs. Parasites & Vectors, 16(1):341, Oct. 2023. ISSN 1756-3305. doi: 10.1186/s13071-023-05956-1. URL https://doi.org/10.1186/s13071-023-05956-1.

JOCHER, G.; QIU, J. Ultralytics yolo11. 2024. URL https://github.com/ultralytics/ultralytics.

JORDAN, M. I.; MITCHELL, T. M. Machine learning: trends, perspectives, and prospects. 2015. URL https://www.science.org/doi/10.1126/science.aaa8415.

KAGGLE. Kaggle. 2025. URL https://www.kaggle.com/. Accessed: 2025-02-19.

KLEINBERG, R.; LI, Y.; YUAN, Y. An alternative view: when does SGD escape local minima? 2018. URL https://arxiv.org/abs/1802.06175.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. ImageNet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C. J. C.; BOTTU, L.; WEINBERGER, K. Q. (eds.) Advances in Neural Information Processing Systems, v. 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

LIGHTNING AI. Lightning AI. 2025. URL https://lightning.ai/. Accessed: 2025-01-05.

LIN, T.-Y.; DOLLÁR, P.; GIRSHICK, R.; HE, K.; HARIHARAN, B.; BELONGIE, S. Feature pyramid networks for object detection. 2017. URL https://arxiv.org/abs/1612.03144.

LOSHCHILLOV, I.; HUTTER, F. Decoupled weight decay regularization. 2019. URL https://arxiv.org/abs/1711.05101.

APACHE MAVEN. Apache Maven. 2025. URL https://maven.apache.org/.

MICROSOFT. Visual Studio Code. 2025. URL https://code.visualstudio.com/.

MEUSE JR., N. O.; PAPAIS, F. L. F.; SILVA, M. G. Thresholding-based computer vision techniques on automatic counting of Aedes mosquitoes eggs. 2021. Department of Electronics, Federal Institute of Education, Science and Technology of Pernambuco, Recife, Brazil. URL mailto:meusejunior@recife.ifpe.edu.br. (Received 00 Month 2021; revised 00 Month 2021).

ONNX. Open Neural Network Exchange (ONNX). 2025. URL https://onnx.ai/.

OPENCV. OpenCV library. 2024. Version 4.10.0. URL https://opencv.org/.

ORACLE. Java SE documentation. 2025. URL https://docs.oracle.com/en/java/javase/.

ORGANIZAÇÃO PAN-AMERICANA DA SAÚDE – OPAS. Alerta epidemiológico: aumento de casos de dengue na região das Américas. 7 out. 2024. URL https://www.paho.org/pt/documentos/alerta-epidemiologico-aumento-casos-dengue-na-regiao-das-americas-7-outubro-2024. Acesso em: 17 fev. 2025.

OTSU, N. A threshold selection method from gray-level histograms. 1979. URL https://web-ext.u-aizu.ac.jp/course/bmclass/documents/otsu1979.pdf.

PARK, J.; KIM, H.; CHOI, Y.; LEE, J. Differentially private sharpness-aware training. 2023. URL https://arxiv.org/abs/2306.05651.

OSHIRO PAULO DA SILVA ALMEIDA, E.; PASSOS, R. A. Curso de identificação de culicídeos de importância médica. 2018. Apostila de treinamento.

POMBO, A. P. M. Aedes aegypti: morfologia, morfometria do ovo, desenvolvimento embrionário e aspectos relacionados à vigilância entomológica no município de São Paulo. December 2016. Tese de Doutorado, Faculdade de Medicina Veterinária e Zootecnia, Universidade de São Paulo, São Paulo. URL https://doi.org/10.11606/T.10.2017.tde-20032017-183203.

REDMON, J.; DIVVALA, S. K.; GIRSHICK, R. B.; FARHADI, A. You only look once: unified, real-time object detection. CoRR, abs/1506.02640, 2015. URL http://arxiv.org/abs/1506.02640.

REN, S.; HE, K.; GIRSHICK, R.; SUN, J. Faster R-CNN: towards real-time object detection with region proposal networks. CoRR, abs/1506.01497, 2015. URL http://arxiv.org/abs/1506.01497.

ULTRALYTICS. YOLO by Ultralytics: the state-of-the-art object detection framework. 2024. URL https://ultralytics.com/. Accessed: 2024-12-31.

WU, Y.; KIRILLOV, A.; MASSA, F.; LO, W.-Y.; GIRSHICK, R. Detectron2. 2019. URL https://github.com/facebookresearch/detectron2.

YU, F.; KOLTUN, V. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122, 2015. doi: 10.48550/arXiv.1511.07122. URL https://arxiv.org/abs/1511.07122.