

# Evaluación Comparativa de YOLOv8, YOLOv11 y YOLOv26 para la Detección Automática de Mosquitos y Larvas de *Aedes aegypti*

## Comparative Evaluation of YOLOv8, YOLOv11, and YOLOv26 for the Automatic Detection of *Aedes aegypti* Mosquitoes and Larvae

Campa-Ramírez, E. R.<sup>1</sup>; González-Vargas, L. A.<sup>2</sup> ✉; Alvarado-Tovar, N.<sup>3</sup>; Vidaña-Martínez, S. A.<sup>4</sup>; Hernández-Rodríguez, R.<sup>5</sup>.

### Datos de Adscripción:

<sup>1</sup> Campa Ramírez, Edson Raúl  
Tecnológico Nacional de México / Instituto Tecnológico de la Laguna.  
División de Estudios de Posgrado e Investigación. Torreón, Coahuila. México.  
ROR ID: <https://ror.org/00davyr38>  
ROR ID: <https://ror.org/00xbw4a70>  
m.ercampar@lalaguna.tecnm.mx  
ORCID: <https://orcid.org/0009-0008-3327-5912>

<sup>2</sup> ✉ González Vargas, Luis Amado  
Tecnológico Nacional de México / Instituto Tecnológico Superior de Lerdo.  
Departamento de Posgrado. Ciudad Lerdo, Durango. México  
Instituto Tecnológico Superior de Lerdo.  
ROR ID: <https://ror.org/04bjtbt12>  
luis.gv@lerdo.tecnm.mx  
ORCID: <https://orcid.org/0009-0008-9094-6188>

<sup>3</sup> Alvarado Tovar, Noé  
Tecnológico Nacional de México / Instituto Tecnológico de la Laguna.  
División de Estudios de Posgrado e Investigación. Torreón, Coahuila. México.  
Tecnológico Nacional de México. ROR ID: <https://ror.org/00davyr38>  
Instituto Tecnológico de la Laguna. ROR ID: <https://ror.org/00xbw4a70>  
nalvarado@lalaguna.tecnm.mx  
ORCID: <https://orcid.org/0000-0002-0218-0813>

<sup>4</sup> Vidaña Martínez, Silvia Adriana  
Tecnológico Nacional de México / Instituto Tecnológico Superior de Lerdo  
División de Ingeniería Ambiental. Ciudad Lerdo, Durango. México.  
Instituto Tecnológico Superior de Lerdo.  
ROR ID: <https://ror.org/04bjtbt12>  
silvia.vm@itslerdo.edu.mx  
ORCID: <https://orcid.org/0009-0008-9633-1010>

<sup>5</sup> Hernández Rodríguez, Rubén  
Tecnológico Nacional de México / ITS de San Pedro de las Colonias.  
División de Ingeniería en Sistemas Computacionales. San Pedro, Coahuila.  
México.  
Instituto Tecnológico Superior de San Pedro de las Colonias.  
ROR ID: <https://ror.org/0134he436>  
rhernandez@tecsanpedro.edu.mx  
ORCID: <https://orcid.org/0000-0002-6903-9988>

**Resumen** - El dengue representa una de las principales amenazas para la salud pública en México, con aproximadamente 125 000 casos y cerca de 480 defunciones registradas en 2024. La Región Lagunera, ubicada en el estado de Coahuila, es una zona vulnerable debido a las condiciones climáticas que favorecen la proliferación del vector *Aedes aegypti*. Los métodos tradicionales de vigilancia entomológica, basados en el conteo manual de mosquitos y larvas, presentan limitaciones en cuanto al tiempo de procesamiento, la susceptibilidad al error humano y la baja capacidad de respuesta oportuna. En este contexto, el presente estudio evalúa y compara el desempeño de las versiones 8, 11 y 26 del algoritmo You Only Look Once para la detección automática de mosquitos adultos y larvas de *Aedes aegypti*. Se construyó un conjunto de datos con dos clases, etiquetado mediante cajas delimitadoras y dividido en subconjuntos de entrenamiento, validación y prueba. Los modelos fueron entrenados bajo condiciones homogéneas y

evaluados mediante métricas de precisión, exhaustividad, medida armónica entre ambas métricas y precisión promedio, además de análisis de matrices de confusión y rendimiento computacional. Los resultados permitieron identificar la arquitectura con el mejor equilibrio entre exactitud de detección y eficiencia computacional, lo que contribuye al desarrollo de herramientas automatizadas para el monitoreo epidemiológico en regiones vulnerables.

**Palabras clave:** *Aedes aegypti*, Aprendizaje profundo, Dengue, Detección de objetos, Vigilancia vectorial, You Only Look Once.

**Abstract** - Dengue is one of the major public health threats in Mexico, with approximately 125,000 reported cases and nearly 480 deaths recorded during 2024. The Laguna Region, in the state of Coahuila, is considered vulnerable because its climatic conditions favor the proliferation of the vector *Aedes aegypti*. Traditional entomological surveillance methods, based on the manual counting of mosquitoes and larvae, present limitations related to processing time, susceptibility to human error, and limited capacity for timely response. In this context, the present study evaluates and compares versions 8, 11, and 26 of the You Only Look Once algorithm for automatic detection of adult mosquitoes and *Aedes aegypti* larvae. We constructed a two-class dataset, annotated with bounding boxes, and divided it into training, validation, and test subsets. The models were trained under homogeneous conditions and evaluated using precision, recall, the harmonic mean of precision and recall, and mean average precision, along with confusion matrix analysis and computational performance assessment. The results identified the architecture that best balances detection accuracy and computational efficiency, contributing to the development of automated tools for epidemiological surveillance in vulnerable regions.

**Keywords:** *Aedes aegypti*, deep learning, dengue, object detection, vector surveillance, You Only Look Once.

### I. INTRODUCCIÓN

El dengue constituye la arbovirosis de mayor impacto a nivel mundial, con aproximadamente 390 millones de infecciones anuales, y representa uno de los principales desafíos para los sistemas de salud pública debido a su rápida expansión geográfica y al incremento sostenido de su incidencia. En México, durante 2024 se registraron alrededor de 125 000 casos

confirmados y 480 defunciones, lo que evidencia una tendencia ascendente de la enfermedad y una creciente vulnerabilidad en entidades como Colima y Jalisco (Cremades et al., 2026). Asimismo, la incidencia nacional aumentó de 29.4 a 279.0 casos por cada 100 000 habitantes entre 2022 y 2024, extendiéndose al 68.6 % de los municipios del país, lo que refleja una rápida dispersión de *Aedes aegypti* (Mendoza-Cano et al., 2025).

En la Región Lagunera del estado de Coahuila, las condiciones climáticas favorecen el establecimiento y proliferación del vector, registrándose recientemente 5 725 casos confirmados. Ante este panorama, la vigilancia entomológica constituye una herramienta fundamental para la detección temprana y el control oportuno de brotes epidémicos. Sin embargo, los métodos tradicionales, basados en la inspección manual de ovitrampas, larvas y mosquitos adultos, siguen siendo procesos laboriosos, dependientes de personal especializado y susceptibles a errores humanos, lo que limita la capacidad de respuesta de los programas de vigilancia epidemiológica (Sasmita et al., 2021).

En los últimos años, la inteligencia artificial y, en particular, el aprendizaje profundo, han demostrado un elevado potencial para automatizar tareas de identificación y clasificación en aplicaciones de visión por computadora. En el ámbito de la vigilancia entomológica, Javed et al. (2023), desarrollaron EggCountAI, un sistema basado en redes neuronales convolucionales capaz de alcanzar exactitudes de hasta 98.88 % en el conteo automático de huevos de *Aedes aegypti*. De manera similar, Duan et al. (2026) propusieron LW-YOLO, una versión ligera de la arquitectura YOLO (You Only Look Once) que redujo en aproximadamente un 60 % el número de parámetros del modelo, manteniendo valores de mAP@50 superiores a 0.97 para la detección de mosquitos adultos. Estos trabajos demuestran la viabilidad técnica del aprendizaje profundo para fortalecer los sistemas automatizados de vigilancia entomológica, especialmente en escenarios con limitaciones computacionales. Dentro de este contexto, la familia YOLO se ha consolidado como una de las arquitecturas de detección de objetos más utilizadas, debido a su capacidad para equilibrar la precisión de detección y la velocidad de inferencia. La evolución de estas arquitecturas, desde YOLOv8 hasta YOLOv11, ha incorporado mejoras en los mecanismos de atención, las estrategias de extracción de características y el escalamiento dinámico, lo que ha incrementado la eficiencia computacional sin afectar significativamente la precisión de la detección. No obstante, estudios recientes indican que las diferencias de desempeño entre generaciones sucesivas suelen ser reducidas cuando se emplean conjuntos de datos específicos, lo que sugiere que las innovaciones actuales están orientadas principalmente a optimizar la implementación en dispositivos de borde y sistemas embebidos, más que a incrementar sustancialmente la exactitud de la clasificación (Hung & Rodríguez, 2025; Hendriko & Hermanto, 2025).

A pesar de estos avances, la mayoría de las investigaciones publicadas se ha centrado en una única etapa del ciclo biológico del mosquito (larvas o adultos), ha empleado conjuntos de datos obtenidos en condiciones altamente controladas o ha evaluado una sola arquitectura de detección. En consecuencia, son escasos los estudios que comparan distintas generaciones de modelos YOLO utilizando un protocolo experimental homogéneo e incorporando simultáneamente indicadores de precisión de detección, de capacidad de localización espacial y de eficiencia computacional. Esta limitación dificulta la selección objetiva de la

arquitectura más adecuada para aplicaciones de vigilancia epidemiológica en regiones endémicas como la Comarca Lagunera, donde las condiciones ambientales representan un desafío adicional para los sistemas de detección automática.

En este contexto, el presente estudio desarrolla una evaluación experimental comparativa del desempeño de YOLOv8, YOLOv11 y YOLOv26 para la detección automática de larvas y mosquitos adultos de *Aedes aegypti* y *Culex*. Para ello, se construyó un conjunto de datos estandarizado con imágenes adquiridas bajo diferentes condiciones de iluminación, orientación, fondo y escala, las cuales fueron etiquetadas mediante bounding boxes y divididas en subconjuntos de entrenamiento (70 %), validación (20 %) y prueba (10 %). Los modelos fueron entrenados bajo un protocolo homogéneo de hiperparámetros y número de épocas, lo que garantiza una comparación objetiva entre arquitecturas. Su desempeño se evaluó mediante métricas ampliamente aceptadas en detección de objetos (Precisión, Recall, F1-score, mAP50, mAP50-95, matrices de confusión), así como mediante análisis de convergencia, estabilidad del entrenamiento y rendimiento computacional (tiempo de inferencia y cuadros por segundo, FPS).

## II. PARTE TÉCNICA DEL ARTÍCULO

### 2.1. Metodología

La presente investigación se desarrolló con un enfoque experimental comparativo, orientado a evaluar el desempeño de tres arquitecturas de detección de objetos basadas en aprendizaje profundo (YOLOv8, YOLOv11 y YOLOv26) en condiciones homogéneas de entrenamiento y validación. El diseño experimental se estructuró siguiendo las recomendaciones para la evaluación objetiva de modelos de visión por computadora, manteniendo constantes los hiperparámetros, la resolución de entrada y los criterios de evaluación, con el fin de minimizar los sesgos durante la comparación (Ultralytics, 2024; Jocher et al., 2024).

La Figura 1 resume el flujo metodológico implementado, que comprende seis etapas principales: adquisición de imágenes RGB, segmentación morfológica, construcción y etiquetado del conjunto de datos, entrenamiento de los modelos, evaluación mediante métricas estandarizadas y selección del modelo con mejor desempeño.

**Figura 1.**  
Esquema experimental comparativo



### 2.2 Materiales y Herramientas Utilizadas

El entrenamiento y evaluación de los modelos se realizó utilizando una infraestructura basada en computación en la nube, lo que permitió garantizar la disponibilidad de recursos GPU durante todas las fases experimentales. Asimismo, se emplearon herramientas ampliamente utilizadas en la investigación en visión por computadora, lo que favoreció la reproducibilidad del estudio

(Google Colab, Ultralytics y Roboflow). La Tabla 1 describe los distintos recursos utilizados.

**Tabla 1**  
Especificaciones de hardware, software y datos.

| Categoría | Detalle                                                 |
|-----------|---------------------------------------------------------|
| Hardware  | GPU NVIDIA Tesla T4 (15 GB RAM), Entorno Google Colab   |
| Software  | Python 3.x, Ultralytics YOLO, PyTorch, MATLAB, Roboflow |
| Datasets  | Larvas (1,190 imágenes), Adultos (1,749 imágenes)       |
| Fuentes   | Kaggle (larvas/adultos) y Roboflow (adultos).           |

La utilización de GPU NVIDIA Tesla T4 permitió reducir considerablemente los tiempos de entrenamiento, mientras que la integración entre Roboflow y Ultralytics facilitó la administración del conjunto de datos y la estandarización del proceso de entrenamiento (Roboflow, 2024; Ultralytics, 2024).

### 2.3 Segmentación Morfológica en MATLAB (Ground Truth)

La segmentación morfológica es una de las técnicas clásicas de procesamiento digital de imágenes para separar objetos del fondo mediante operadores matemáticos aplicados a imágenes binarias. Este tipo de procesamiento continúa utilizándose como etapa previa para la generación de máscaras de referencia (ground truth) destinadas al entrenamiento de modelos supervisados (González & Woods, 2018). El proceso consistió en las siguientes etapas: conversión a escala de grises, suavizado mediante filtro Gaussiano, que reduce el ruido de alta frecuencia preservando las estructuras principales de la imagen, favoreciendo posteriormente la detección de bordes mediante operadores diferenciales como Sobel (González & Woods, 2018) mientras que método de Otsu selecciona automáticamente el umbral óptimo maximizando la varianza entre clases, siendo ampliamente utilizado en segmentación automática de imágenes biomédicas y entomológicas (Otsu, 1979), las operaciones morfológicas de cierre permiten eliminar discontinuidades y consolidar regiones conectadas, mejorando la definición de los contornos empleados para la generación de máscaras binarias. Este procedimiento se aplicó tanto a imágenes del género *Aedes aegypti* como *Culex*, generando conjuntos de datos más limpios y consistentes que sirvieron como base para el etiquetado. La Figura 2 muestra la segmentación aplicada a un espécimen representativo del género *Aedes*, de izquierda a derecha: imagen original en escala de grises, detección con Sobel, umbral inicial, cierre morfológico, máscara final y segmentación resultante. Por otra parte, la Figura 3 muestra algunos ejemplos del resultado final de la segmentación de ambas especies.

**Figura 2**  
Segmentación morfológica en MATLAB aplicado a mosquito *Aedes*

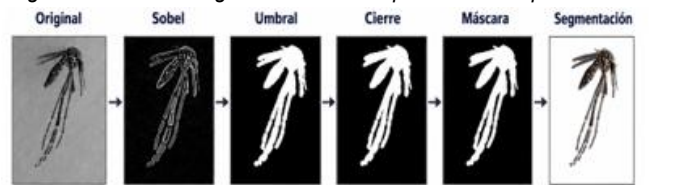


**Figura 3**  
Ejemplos de mosquitos segmentados del tipo *Aedes* y *Culex*



### 2.4 Generación de dataset

La calidad del conjunto de datos es uno de los factores de mayor influencia en el desempeño de los modelos de aprendizaje profundo. Diversos estudios han demostrado que la diversidad en la iluminación, la orientación y la escala favorece la capacidad de generalización en la inferencia (Shorten & Khoshgoftaar, 2019). El dataset fue gestionado mediante la plataforma Roboflow. El primer conjunto estuvo conformado por 1,190 imágenes de larvas (clases: Larva *Aedes* y Larva *Culex*), mientras que el segundo incluyó 1,749 imágenes de mosquitos adultos (clases: *Aedes* y *Culex*). Las imágenes fueron anotadas mediante la técnica de bounding boxes (cajas delimitadoras), que es el estándar internacional para tareas de detección de objetos, ya que permite localizar espacialmente cada instancia y asociarla con su categoría correspondiente durante el entrenamiento supervisado (Redmon et al., 2016). El etiquetado permitió generar las anotaciones necesarias para el entrenamiento supervisado de los modelos YOLO, lo que facilitó la localización y la clasificación automáticas de los organismos durante la inferencia. Los conjuntos de datos fueron particionados en 70% para entrenamiento, 20% para validación y 10% para prueba. Esta distribución permitió entrenar los modelos y evaluar su desempeño en datos no vistos previamente, como se muestra en la Tabla 2.

**Tabla 2**  
Distribución de imágenes por conjunto de datos.

| Conjunto de datos | Entrenamiento | Validación | Prueba   | Total |
|-------------------|---------------|------------|----------|-------|
| Larvas            | 833 (1,041)   | 238 (99)   | 119 (50) | 1,190 |
| Mosquitos adultos | 1,224 (1,530) | 350 (146)  | 175 (73) | 1,749 |

Con el objetivo de incrementar la variabilidad de las muestras y mejorar la capacidad de generalización de los modelos, se aplicaron técnicas de aumento de datos (data augmentation), que reducen el sobreajuste e incrementan la robustez del modelo al simular variaciones presentes durante la adquisición real de imágenes (Shorten & Khoshgoftaar, 2019). En el caso de las larvas, se utilizó una aumentación de  $\times 3$  mediante volteos horizontales y verticales, rotaciones de  $\pm 9^\circ$  y variaciones de brillo de  $\pm 17\%$ . Estas transformaciones permitieron simular diferentes orientaciones y condiciones de iluminación presentes durante la captura microscópica de las muestras.

Para los mosquitos adultos se aplicaron rotaciones de  $\pm 15^\circ$  y ajustes de brillo de  $\pm 15\%$ , excluyendo los volteos horizontales y verticales. Esta decisión se tomó porque la orientación corporal y la disposición anatómica son características relevantes para la diferenciación taxonómica entre especies, por lo que un giro podría alterar patrones morfológicos importantes para el aprendizaje del modelo.

## 2.5 Entrenamiento de los Modelos

Las arquitecturas YOLO pertenecen a la familia de detectores de una sola etapa (one-stage detectors), caracterizadas por combinar velocidad de inferencia y alta precisión, atributos que las convierten en una alternativa adecuada para aplicaciones de monitoreo en tiempo real (Redmon et al., 2016; Jocher et al., 2024).

El entrenamiento de los modelos se realizó en Python mediante la librería Ultralytics YOLO sobre el entorno de ejecución de Google Colab con aceleración mediante GPU. Se evaluaron las arquitecturas YOLOv8, YOLOv11 y YOLOv26 en sus variantes Nano (n), utilizando pesos preentrenados en formato PyTorch (.pt). Las versiones Nano fueron seleccionadas por presentar una menor complejidad computacional sin comprometer significativamente la precisión, lo cual resulta relevante para futuras implementaciones en plataformas embebidas como Raspberry Pi o NVIDIA Jetson (Ultralytics, 2024). Los conjuntos de datos fueron gestionados con Roboflow y exportados en formato compatible con Ultralytics mediante archivos de configuración YAML. Durante el entrenamiento se mantuvieron condiciones homogéneas en los experimentos, con el objetivo de garantizar una comparación objetiva entre arquitecturas.

La utilización de pesos preentrenados permitió aprovechar las ventajas del aprendizaje por transferencia (transfer learning), una estrategia ampliamente utilizada para acelerar la convergencia y mejorar el desempeño cuando se dispone de conjuntos de datos especializados (Pan & Yang, 2010).

La comparación entre arquitecturas se realizó exactamente bajo las mismas condiciones experimentales, evitando introducir factores externos que pudieran favorecer el desempeño de alguno de los modelos evaluados.

## 2.6 Evaluación

La evaluación del desempeño se realizó utilizando las métricas recomendadas para problemas de detección de objetos, que permiten evaluar simultáneamente la capacidad de clasificación y la precisión espacial de las detecciones (Everingham et al., 2015).

El desempeño de los modelos fue evaluado mediante cinco métricas estándar de detección de objetos, calculadas automáticamente por la librería Ultralytics durante el proceso de validación.

La Precisión (P), se calcula mediante (1) y mide la proporción de detecciones correctas (verdaderos positivos, TP) sobre el total de detecciones realizadas, penalizando los falsos positivos (FP):

$$\text{Precisión} = \frac{TP}{TP+FP} \quad (1)$$

El Recall (R), expresado en (2), indica la proporción de objetos reales que el modelo detectó correctamente, penalizando los falsos negativos (FN):

$$\text{Recall} = \frac{TP}{TP+FN} \quad (2)$$

El F1-score es la media armónica entre Precisión y Recall, ofreciendo una métrica balanceada (3):

$$\text{F1-score} = 2 \frac{P \cdot R}{P+R} \quad (3)$$

La Precisión Promedio (AP), obtenida mediante (4), integra el área bajo la curva Precisión-Recall a través de diferentes umbrales de confianza:

$$AP = \sum_k (R_k - R_{k-1}) \cdot P_k \quad (4)$$

La media de la Precisión Promedio (mAP), expresada en (5), promedia AP sobre todas las clases  $N$  del conjunto de datos:

$$\text{mAP} = \frac{1}{N} \cdot \sum AP_i \quad (5)$$

La intersección sobre la unión de objetos (IoU), expresada en (6), determina el grado de superposición entre la caja predicha y el ground truth:

$$\text{IoU} = \frac{\text{Área(Predicción intersección GT)}}{\text{Área (Predicción unión GT)}} \quad (6)$$

El IoU constituye el criterio internacional utilizado para determinar el grado de coincidencia entre la predicción del modelo y la anotación de referencia (ground truth), y sirve de base para el cálculo del Average Precision y del mean Average Precision (Everingham et al., 2015).

En el mAP50, el umbral de IoU se fija en 0.50. El mAP50-95 calcula el mAP de forma progresiva a partir de umbrales de IoU desde 0.50 hasta 0.95, con incrementos de 0.05, lo que proporciona una evaluación más rigurosa de la precisión de localización. Esta última métrica se utilizó como criterio principal de comparación entre arquitecturas en el presente estudio. Actualmente, la métrica mAP50-95 se considera el estándar de referencia para comparar arquitecturas de detección de objetos, ya que evalúa simultáneamente la precisión de localización a diferentes niveles de exigencia (Lin et al., 2014; Jocher et al., 2024).

Se evaluaron distintas configuraciones de épocas con el objetivo de analizar la convergencia y desempeño de cada modelo. Para cada versión de YOLO, el entrenamiento se realizó en incrementos de 50 épocas, hasta un máximo de 300 épocas. Durante el proceso, Ultralytics almacenó automáticamente los pesos del mejor modelo obtenido en cada experimento, utilizando como criterio principal la métrica mAP50-95 calculada sobre el conjunto de validación. Los modelos resultantes fueron guardados en formato PyTorch (.pt), lo que permitió posteriormente su evaluación e inferencia para analizar los resultados. El entrenamiento se realizó bajo condiciones homogéneas para asegurar una comparación justa entre arquitecturas, manteniendo constantes parámetros como la resolución de entrada, la configuración de los datos y las métricas de evaluación. Estas métricas fueron calculadas automáticamente mediante la librería Ultralytics YOLO durante el proceso de validación de cada modelo.

La Tabla 3 presenta el desempeño de los modelos YOLOv8, YOLOv11 y YOLOv26 evaluados con diferentes configuraciones del número de épocas durante el entrenamiento. El objetivo de esta comparación es analizar la evolución del rendimiento de cada modelo en función del número de iteraciones e identificar el punto de mejor desempeño.

**Tabla 3**  
Métricas por época — Modelos YOLO, larvas Aedes y Culex.

| Modelo  | Épocas | Precisión (%) | Recall (%) | F1-score (%) | mAP50 (%)    | mAP50-95 (%) |
|---------|--------|---------------|------------|--------------|--------------|--------------|
| YOLOv8  | 50     | 99.34         | 100.00     | 99.67        | <b>99.50</b> | 78.34        |
| YOLOv8  | 100    | 99.84         | 100.00     | 99.92        | <b>99.50</b> | 79.14        |
| YOLOv8  | 150    | 99.84         | 100.00     | 99.92        | <b>99.50</b> | 78.84        |
| YOLOv8  | 250    | 99.72         | 100.00     | 99.86        | <b>99.50</b> | 79.00        |
| YOLOv8  | 257    | 98.74         | 99.06      | 98.90        | 99.44        | 78.60        |
| YOLOv11 | 50     | 98.84         | 100.00     | 99.41        | 99.20        | 79.13        |
| YOLOv11 | 100    | 99.79         | 100.00     | 99.89        | <b>99.50</b> | 78.73        |
| YOLOv11 | 150    | 99.74         | 100.00     | 99.87        | <b>99.50</b> | <b>80.34</b> |
| YOLOv11 | 200    | 99.67         | 100.00     | 99.83        | <b>99.50</b> | 80.27        |
| YOLOv11 | 250    | 99.27         | 100.00     | 99.63        | <b>99.50</b> | 79.56        |
| YOLOv11 | 300    | 99.54         | 100.00     | 99.77        | <b>99.50</b> | 78.79        |
| YOLOv26 | 50     | 100.00        | 99.48      | 99.74        | <b>99.50</b> | 80.24        |
| YOLOv26 | 100    | 99.83         | 100.00     | 99.91        | <b>99.50</b> | 79.34        |
| YOLOv26 | 150    | 99.85         | 99.92      | 99.88        | <b>99.50</b> | 79.51        |
| YOLOv26 | 200    | 99.19         | 100.00     | 99.59        | <b>99.50</b> | 80.28        |
| YOLOv26 | 250    | 99.87         | 100.00     | 99.93        | <b>99.50</b> | 79.43        |
| YOLOv26 | 300    | 99.69         | 100.00     | 99.84        | <b>99.50</b> | 77.55        |
| YOLOv26 | 350    | 98.83         | 100.00     | 99.41        | 99.36        | 78.46        |
| YOLOv26 | 363    | 99.76         | 100.00     | 99.88        | <b>99.50</b> | 78.26        |

Nota. Los valores en negrita indican el mejor mAP50-95.

En el caso de YOLOv8, se evaluaron configuraciones de 50 a 257 épocas. Se observa que el modelo alcanza valores elevados de precisión y recall desde etapas tempranas del entrenamiento, manteniendo un recall del 100% en la mayoría de las configuraciones. El mejor desempeño en términos de mAP50-95 se obtuvo alrededor de las 100 épocas, con un valor de 79.14%, lo que indica una convergencia relativamente rápida.

Por su parte, YOLOv11 se evaluó en un rango de 50 a 300 épocas. Este modelo mantiene un comportamiento estable, con valores de precisión cercanos al 99.7% y de recall del 100% en todas las configuraciones. El mejor resultado se obtuvo a las 150 épocas, con un mAP50-95 de 80.34%, lo que representa el valor más alto entre todos los modelos evaluados e indica una mayor capacidad de localización precisa.

En el caso de YOLOv26, se evaluaron configuraciones más extensas, de 50 a 363 épocas. Aunque este modelo alcanza valores altos de precisión e incluso el máximo (100%) en algunas configuraciones, su desempeño en mAP50-95 es más variable. El mejor resultado se obtuvo alrededor de las 200 épocas, con un valor de 80.28%, aunque se observa una ligera disminución del rendimiento en entrenamientos más prolongados, lo que sugiere una posible tendencia al sobreajuste.

En general, los tres modelos presentan un desempeño sobresaliente en términos de mAP50, con valores cercanos al 99.50% en todos los casos, y un recall cercano al 100%, lo que indica una alta capacidad de detección. Sin embargo, la métrica mAP50-95 permite distinguir con mayor precisión el rendimiento real, evidenciando que YOLOv11 ofrece el mejor equilibrio entre precisión y estabilidad durante el entrenamiento. Finalmente, el

rendimiento de los modelos se evaluó mediante las métricas de precisión, recall, F1-score, mAP50 y mAP50-95, lo que permitió comparar cuantitativamente las distintas arquitecturas.

**Tabla 4**  
Mejores métricas por modelo — Larvas Aedes y Culex.

| Modelo  | Épocas | P (%) | R (%)  | F1 (%) | mAP50 (%)    | mAP50-95 (%) |
|---------|--------|-------|--------|--------|--------------|--------------|
| YOLOv8  | 100    | 99.84 | 100.00 | 99.92  | <b>99.50</b> | 79.14        |
| YOLOv11 | 150    | 99.74 | 100.00 | 99.87  | <b>99.50</b> | <b>80.34</b> |
| YOLOv26 | 200    | 99.19 | 100.00 | 99.59  | <b>99.50</b> | 80.28        |

Nota. Mejor resultado por modelo según mAP50-95. Negrita = mejor por columna.

De acuerdo con la Tabla 3, los tres modelos alcanzaron Recall del 100% y mAP50 de 99.50%. YOLOv11 a 150 épocas obtuvo el mayor mAP50-95 con 80.34%, seguido de YOLOv26 (80.28%) y YOLOv8 (79.14%). YOLOv8 destacó por convergencia rápida desde las 100 épocas. YOLOv26 mostró tendencia al sobreajuste en configuraciones superiores a 300 épocas. La Tabla 5 presenta los resultados obtenidos para la detección de mosquitos adultos de los géneros Aedes y Culex, evaluando el desempeño de los modelos YOLOv8, YOLOv11 y YOLOv26 bajo distintas configuraciones de épocas de entrenamiento.

**Tabla 5**  
Métricas de entrenamiento por época — Modelos YOLO, dataset de mosquitos adultos Aedes y Culex.

| Modelo  | Épocas | Precisión (%) | Recall (%) | F1-score (%) | mAP50 (%)    | mAP50-95 (%) |
|---------|--------|---------------|------------|--------------|--------------|--------------|
| YOLOv8  | 50     | 99.88         | 99.56      | 99.72        | <b>99.50</b> | 77.07        |
| YOLOv8  | 100    | 99.73         | 98.53      | 99.13        | 98.50        | 76.29        |
| YOLOv8  | 150    | 99.76         | 100.00     | 99.88        | <b>99.50</b> | 76.48        |
| YOLOv8  | 174    | 99.76         | 98.53      | 99.14        | 98.50        | 75.65        |
| YOLOv8  | 185    | 99.77         | 98.53      | 99.15        | 99.42        | 75.74        |
| YOLOv8  | 208    | 99.82         | 98.53      | 99.17        | 98.50        | 74.90        |
| YOLOv11 | 50     | 99.85         | 100.00     | 99.92        | <b>99.50</b> | 77.52        |
| YOLOv11 | 100    | 99.48         | 98.53      | 99.00        | 98.50        | 77.07        |
| YOLOv11 | 150    | 99.81         | 98.41      | 99.10        | 98.50        | <b>78.63</b> |
| YOLOv11 | 200    | 99.76         | 98.53      | 99.14        | 98.50        | 77.44        |
| YOLOv11 | 250    | 99.70         | 98.53      | 99.11        | 98.50        | 78.97        |
| YOLOv11 | 292    | 99.44         | 98.53      | 98.98        | 98.50        | 77.13        |
| YOLOv26 | 50     | 98.58         | 99.55      | 99.06        | <b>99.50</b> | 78.21        |
| YOLOv26 | 100    | 99.23         | 99.90      | 99.56        | <b>99.50</b> | 78.38        |
| YOLOv26 | 150    | 99.95         | 98.40      | 99.17        | 98.50        | 77.14        |
| YOLOv26 | 208    | 99.47         | 97.82      | 98.64        | 98.50        | 75.87        |
| YOLOv26 | 248    | 98.44         | 98.17      | 98.30        | 98.41        | 77.61        |

Los valores en negrita indican el mejor mAP50-95 global. Las líneas separan los grupos por modelo.

Las métricas de entrenamiento evidencian que las tres arquitecturas evaluadas desarrollaron un proceso de aprendizaje estable y consistente para la detección de mosquitos adultos de Aedes y Culex. En la mayoría de las configuraciones analizadas,

los modelos alcanzaron valores de Precisión y F1-score superiores al 98%, lo que demuestra una elevada capacidad para identificar correctamente los especímenes y mantener un equilibrio adecuado entre falsos positivos y falsos negativos. Asimismo, los valores de mAP50, cercanos al 99.50%, reflejan una alta capacidad de localización bajo un criterio de coincidencia espacial moderado ( $IoU = 0.50$ ).

A diferencia del conjunto de datos de larvas, el recall presentó una ligera disminución, alcanzando un máximo de 98.53%. Este comportamiento puede atribuirse, por una parte, a la mayor complejidad morfológica de los mosquitos adultos, caracterizados por una mayor variabilidad en la orientación, la postura corporal, la escala y la iluminación durante la adquisición de las imágenes. Adicionalmente, el desbalance entre las clases evaluadas en el conjunto de prueba (112 especímenes de *Aedes* frente a 34 de *Culex*), pudo influir parcialmente en la capacidad del modelo para recuperar la totalidad de los individuos presentes, afectando el valor del recall.

La evaluación mediante mAP50-95, considerada una de las métricas más estrictas para comparar modelos de detección de objetos, permitió diferenciar con mayor claridad el desempeño de las arquitecturas evaluadas. En este sentido, de acuerdo con la Tabla 6, YOLOv11 obtuvo el mejor rendimiento, alcanzando un mAP50-95 de 78.97 % tras 250 épocas de entrenamiento, seguido por YOLOv26, con 78.38 % a las 100 épocas, y YOLOv8, con 77.07 % a las 50 épocas. Aunque las diferencias entre los modelos fueron relativamente pequeñas, estos resultados evidencian una mayor estabilidad y capacidad de localización espacial en YOLOv11, mientras que YOLOv8 destacó por alcanzar su desempeño óptimo en un menor número de épocas, lo que refleja una convergencia más rápida y una mayor eficiencia computacional. Por su parte, YOLOv26 presentó un comportamiento competitivo en las primeras etapas del entrenamiento; sin embargo, mostró una mayor variabilidad a medida que aumentaba el número de épocas, lo que sugiere una mayor susceptibilidad al sobreajuste.

**Tabla 6**

*Mejores métricas por modelo — Mosquitos adultos Aedes y Culex.*

| Modelo  | Épocas | P (%) | R (%) | F1 (%) | mAP50 (%)    | mAP50-95 (%) |
|---------|--------|-------|-------|--------|--------------|--------------|
| YOLOv8  | 50     | 99.88 | 99.56 | 99.72  | <b>99.50</b> | 77.07        |
| YOLOv11 | 250    | 99.70 | 98.53 | 99.11  | 98.50        | <b>78.97</b> |
| YOLOv26 | 100    | 99.23 | 99.90 | 99.56  | <b>99.50</b> | 78.38        |

Las métricas obtenidas para el conjunto de mosquitos adultos fueron ligeramente inferiores a las alcanzadas en el conjunto de larvas, lo que confirma que la detección de individuos adultos representa un problema de mayor complejidad debido a la diversidad morfológica y a las condiciones de adquisición de las imágenes. En conjunto, estos resultados indican que las tres arquitecturas constituyen alternativas viables para aplicaciones de vigilancia entomológica automatizada; no obstante, YOLOv11 ofrece el mejor equilibrio entre precisión de detección, capacidad de localización espacial y estabilidad durante el entrenamiento, lo que la consolida como la arquitectura más adecuada para este conjunto de datos.

## 2.7 Matrices de Confusión

El análisis de las matrices de confusión para YOLOv11 reveló una clasificación perfecta (1.00) en larvas tras 150 épocas, como se muestra en la Figura 4. En el caso de los mosquitos adultos a 250 épocas, el modelo demostró alta robustez, con 112 detecciones correctas de *Aedes* y 34 de *Culex*, registrando un solo falso positivo, tal como se observa en la Figura 5.

**Figura 4**

*Matriz de confusión normalizada — YOLOv11, 150 épocas, larvas Aedes y Culex. Valores 1.00 indican clasificación perfecta.*

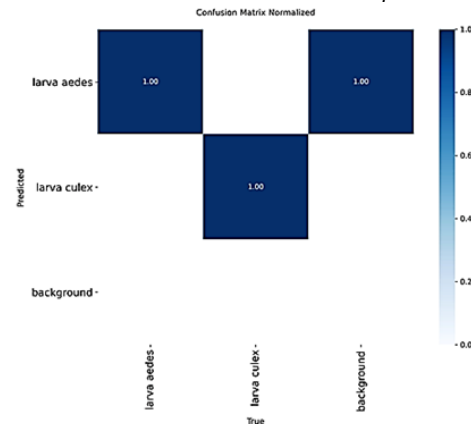
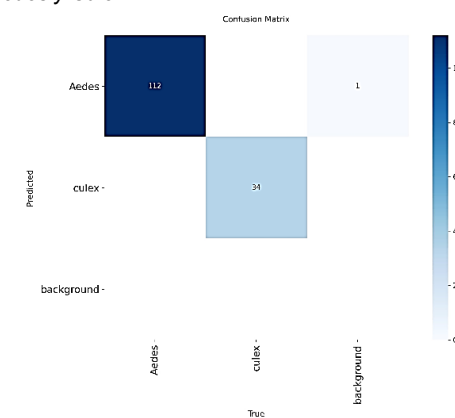


**Figura 5**

*Matriz de confusión absoluta — YOLOv11, 250 épocas, mosquitos adultos Aedes y Culex.*



## III. RESULTADOS Y DISCUSIÓN

Como se observa en las Tablas 3 y 4, las tres arquitecturas alcanzaron un desempeño sobresaliente en la detección de larvas, registrando valores de precisión superiores al 99% y un Recall cercano al 100% en prácticamente todas las configuraciones evaluadas. Estos resultados evidencian que el conjunto de datos empleado permitió un aprendizaje estable e independiente de la arquitectura empleada.

### 3.1 Detección de Larvas de Aedes y Culex

Se observó que YOLOv8 alcanzó su desempeño óptimo tras 100 épocas de entrenamiento, mientras que YOLOv11 y YOLOv26 requirieron un mayor número de iteraciones para alcanzar su máxima capacidad de detección. Esta convergencia

más rápida representa una ventaja desde el punto de vista computacional, ya que reduce el tiempo de entrenamiento y el consumo de recursos, lo que favorece su implementación en escenarios donde la capacidad de procesamiento o la disponibilidad de hardware son limitadas.

### 3.2 Detección de Mosquitos Adultos

Los resultados obtenidos para el conjunto de datos de mosquitos adultos mostraron un comportamiento consistente con el observado en el conjunto de larvas, aunque con una ligera disminución en las métricas de desempeño. Esta diferencia puede atribuirse a la mayor complejidad morfológica de los especímenes adultos, así como a la variabilidad en su orientación, postura corporal, tamaño aparente y condiciones de iluminación durante la adquisición de las imágenes, factores que incrementan la dificultad del proceso de detección y localización automáticos.

En este escenario, YOLOv11 volvió a mostrar el mejor desempeño global, alcanzando un mAP50-95 de 78.97 %, lo que confirma su estabilidad y capacidad de generalización en ambas tareas de detección. YOLOv26 obtuvo un rendimiento muy cercano (78.38 %), aunque mostró una mayor sensibilidad al incremento del número de épocas de entrenamiento, lo que evidencia una tendencia al sobreajuste en las configuraciones de entrenamiento más prolongadas. Por su parte, YOLOv8 destacó por su rápida convergencia, alcanzando un desempeño competitivo con un menor número de épocas, lo que lo convierte en una alternativa adecuada para aplicaciones en las que el tiempo de entrenamiento y los recursos computacionales constituyen factores limitantes.

Las Figuras 6 a 11 ilustran ejemplos representativos de la capacidad de detección alcanzada por YOLOv11 sobre ambos conjuntos de datos. En todos los casos se observa una adecuada localización espacial de los especímenes mediante cajas delimitadoras, acompañada por elevados niveles de confianza. La ausencia de confusiones entre las clases Aedes y Culex coincide con el comportamiento observado en las matrices de confusión y respalda las métricas cuantitativas obtenidas durante la validación.

**Figura 6**  
Ejemplos de detección del modelo de larvas Aedes y Culex

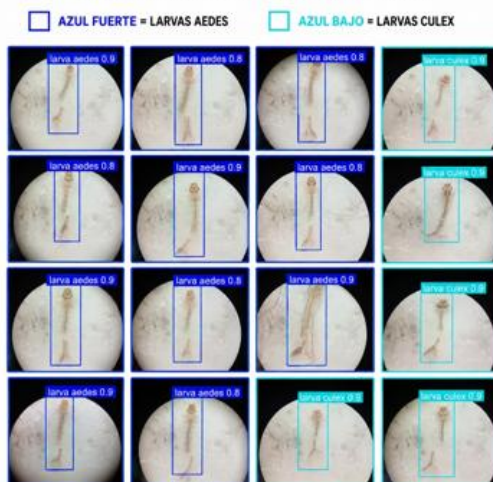


**Figura 7**  
Ejemplos de detección del modelo de mosquitos Aedes y Culex

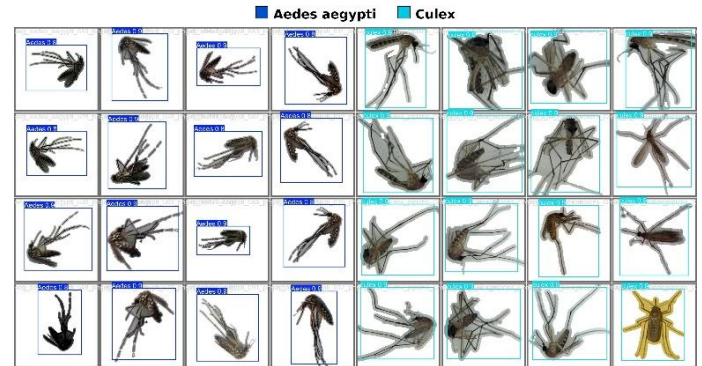


**Figura 8**  
Ejemplo detección con confianza de 0.80

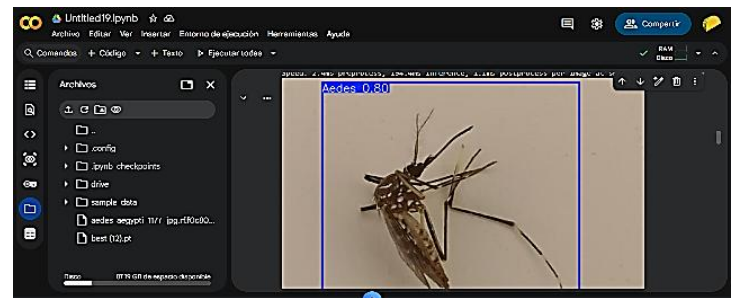


**Figura 9**  
Ejemplo detección con confianza de 0.89

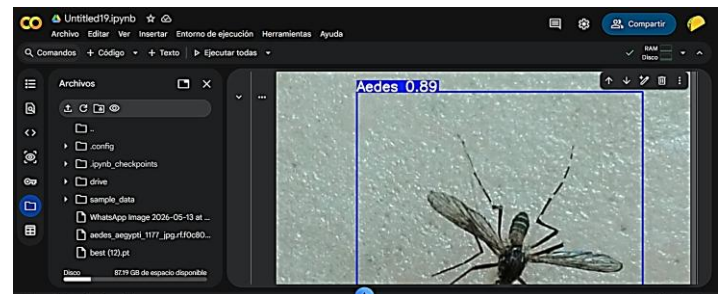
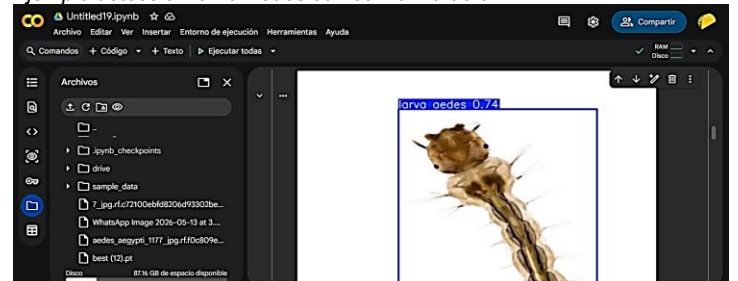


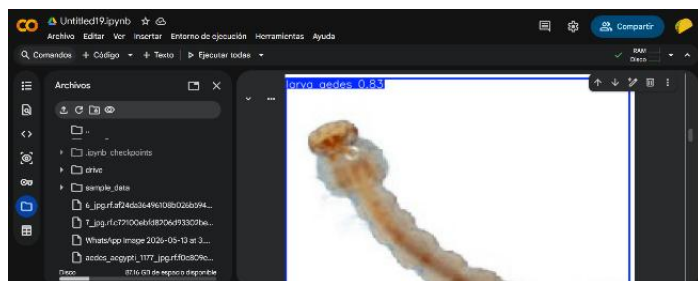
**Figura 10.**  
Ejemplo detección larva Aedes con confianza de 0.74



Otro ejemplo de detección de Larva Aedes con confianza de 0.83, se muestra en la Figura 11.

**Figura 11**

*Ejemplo detección larva Aedes con confianza de 0.83*



Las diferencias observadas entre los conjuntos de larvas y adultos indican que la etapa biológica del organismo influye directamente sobre el desempeño del modelo.

### 3.3 Análisis Comparativo General

Los resultados obtenidos evidencian que las tres arquitecturas evaluadas presentan un desempeño altamente competitivo para la detección automática de larvas y mosquitos adultos de *Aedes aegypti* y *Culex*. Aunque las diferencias fueron mínimas cuando se analizaron métricas tradicionales como precisión, recall y mAP50, la métrica mAP50-95 permitió identificar diferencias más significativas en la precisión de localización espacial, posicionando de manera consistente a YOLOv11 como el modelo con mejor desempeño global. Este comportamiento confirma que el empleo de mAP50-95 constituye un criterio más riguroso y representativo para comparar arquitecturas modernas de detección de objetos, al evaluar el rendimiento bajo diferentes umbrales de IoU.

En contraste, YOLOv8 destacó por alcanzar una convergencia más rápida, obteniendo resultados competitivos con un menor número de épocas de entrenamiento, lo que representa una alternativa atractiva para aplicaciones con restricciones de tiempo o capacidad computacional. Por su parte, YOLOv26 mostró un desempeño comparable al de YOLOv11 en configuraciones intermedias; sin embargo, presentó una mayor tendencia al sobreajuste durante entrenamientos prolongados, lo que sugiere la necesidad de incorporar estrategias de regularización o criterios de parada temprana (early stopping) para preservar su capacidad de generalización.

El análisis de las matrices de confusión respalda los resultados cuantitativos obtenidos. En el conjunto de datos de larvas, YOLOv11 alcanzó una clasificación perfecta, evidenciando una adecuada separación entre las clases y confirmando tanto la consistencia del proceso de entrenamiento como la calidad del conjunto de datos utilizado. Para el conjunto de mosquitos adultos únicamente se registró un falso positivo, lo que demuestra la robustez del modelo incluso frente a una mayor variabilidad morfológica y condiciones experimentales más complejas.

Los hallazgos obtenidos son consistentes con investigaciones recientes sobre detección automática de insectos mediante arquitecturas YOLO. Al Maruf et al. (2025) reportaron elevados niveles de precisión en la clasificación de especies de mosquitos utilizando modelos basados en aprendizaje profundo, mientras que Duan et al. (2026) demostraron que versiones optimizadas

de YOLO pueden mejorar simultáneamente la precisión y la eficiencia computacional. De manera similar, Raza et al. (2026) observaron un desempeño robusto de variantes de YOLO en la detección de larvas bajo diferentes condiciones de adquisición de imágenes. Aunque los valores de mAP50 reportados en estos estudios son, en algunos casos, superiores al 95%, las diferencias pueden atribuirse a la utilización de conjuntos de datos obtenidos en condiciones más controladas y con menor variabilidad ambiental que las consideradas en la presente investigación. En conjunto, los resultados confirman que las tres arquitecturas constituyen alternativas viables para sistemas automatizados de vigilancia entomológica; sin embargo, YOLOv11 ofrece el mejor equilibrio entre precisión de detección, capacidad de localización espacial y estabilidad durante el entrenamiento, lo que la posiciona como la opción más adecuada para futuras implementaciones en plataformas de monitoreo epidemiológico en tiempo real.

## IV. CONCLUSIONES

La presente investigación realizó una evaluación comparativa de las arquitecturas YOLOv8, YOLOv11 y YOLOv26 para la detección automática de larvas y mosquitos adultos de *Aedes aegypti* y *Culex*, mediante un protocolo experimental homogéneo que garantizó una comparación objetiva entre modelos. Los resultados demostraron que las tres arquitecturas presentan un desempeño sobresaliente para aplicaciones de vigilancia entomológica automatizada, alcanzando valores de precisión superiores al 99 %, un recall cercano al 100% y un mAP50 superior al 98.5 %.

Entre los modelos evaluados, YOLOv11 mostró el mejor equilibrio entre precisión de detección, capacidad de localización espacial y estabilidad durante el entrenamiento, obteniendo los mayores valores de mAP50-95 en ambas tareas: 80.34% para la detección de larvas (150 épocas) y 78.97% para la detección de mosquitos adultos (250 épocas). Aunque las diferencias respecto a YOLOv26 fueron reducidas, los resultados evidencian una mayor capacidad de generalización bajo las condiciones experimentales consideradas. Por su parte, YOLOv8 destacó por alcanzar una convergencia más rápida, logrando su desempeño óptimo entre 50 y 100 épocas, lo que representa una alternativa atractiva cuando se dispone de recursos computacionales limitados. En contraste, YOLOv26 mostró un comportamiento competitivo, aunque con una mayor susceptibilidad al sobreajuste durante entrenamientos prolongados.

Desde una perspectiva aplicada, los hallazgos obtenidos respaldan el uso de modelos basados en aprendizaje profundo como herramientas para fortalecer los programas de vigilancia epidemiológica, al reducir la dependencia de procesos manuales, incrementar la eficiencia en la identificación de vectores y favorecer una respuesta más oportuna ante posibles brotes de enfermedades transmitidas por mosquitos. Asimismo, el protocolo metodológico desarrollado constituye una base sólida para futuras investigaciones orientadas a la detección automática de otros vectores de importancia médica y al desarrollo de sistemas inteligentes para la vigilancia sanitaria.

Como líneas futuras de investigación, se propone implementar YOLOv11 en plataformas embebidas, como Raspberry Pi 5, para el monitoreo en tiempo real bajo condiciones de campo; ampliar el conjunto de datos incorporando un mayor número de especies y escenarios ambientales; evaluar arquitecturas emergentes de

detección de objetos; e integrar sistemas de inferencia en tiempo real con infraestructuras IoT y plataformas de vigilancia epidemiológica. Estas acciones contribuirán al desarrollo de soluciones tecnológicas más robustas, escalables y aplicables directamente a programas de salud pública y control vectorial.

## V. AGRADECIMIENTOS

Los autores agradecen al Tecnológico Nacional de México por el apoyo brindado para el desarrollo de esta investigación, así como al Instituto Tecnológico de La Laguna, Instituto Tecnológico Superior de Lerdo e Instituto Tecnológico Superior de San Pedro de las Colonias por las facilidades otorgadas para la realización del proyecto. Asimismo, se reconoce el apoyo de la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) por el impulso al desarrollo científico y tecnológico.

## VI. REFERENCIAS

- Al Maruf, A., Mahmudul Haque, M., Ara Romy, R., Jahan Puspo, J., & Aung, Z. (2025). TransembleNet: Enhancing vector mosquito species classification through transfer learning-based ensemble model. *PLOS ONE*, 20(5), e0322171. <https://doi.org/10.1371/journal.pone.0322171>
- Cremades, R., Sandoval-Pinto, E., Ortega-Prieto, A. M., Jimenez-Guardeño, J. M., Pérez-Gómez, H. R., Lona Reyes, J. C., Sierra-Díaz, E., & Regla-Nava, J. A. (2026). Epidemiological characteristics of dengue disease in Mexico (2014–2025): A descriptive analysis of a hyperendemic country. *Pathogens*, 15(2), 190. <https://doi.org/10.3390/pathogens15020190>
- Duan, J., Liu, L., Ma, D., Kong, M., Wang, S., & Zhu, L. (2026). LW-YOLO: A lightweight enhanced YOLOv8-based model for mosquito detection. *Algorithms*, 19(4), 267. <https://doi.org/10.3390/a19040267>
- Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2015). The Pascal Visual Object Classes Challenge: A retrospective. *International Journal of Computer Vision*, 111(1), 98–136. <https://doi.org/10.1007/s11263-014-0733-5>
- González, R. C., & Woods, R. E. (2018). *Digital image processing* (4th ed.). Pearson.
- Hendriko, V., & Hermanto, D. (2025). Performance comparison of YOLOv10, YOLOv11, and YOLOv12 models on human detection datasets. *Brilliance: Research of Artificial Intelligence*, 5(1), 440–450. <https://doi.org/10.47709/brilliance.v5i1.6447>
- Hung, G., & Rodríguez, I. F. (2025). A comparative study of YOLOv8 to YOLOv11 performance in underwater vision tasks. *arXiv*. <https://doi.org/10.48550/arXiv.2509.12682>
- Javed, N., López-Denman, A. J., Paradkar, P. N., & Bhatti, A. (2023). EggCountAI: A convolutional neural network-based software for counting of *Aedes aegypti* mosquito eggs. *Parasites & Vectors*, 16, 341. <https://doi.org/10.1186/s13071-023-05956-1>
- Jocher, G., Qiu, J., & Chaurasia, A. (2024). *Ultralytics YOLO* (Version 8.2.69) [Computer software]. Zenodo. <https://doi.org/10.5281/zenodo.13129849>
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer vision – ECCV 2014* (pp. 740–755). Springer. [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
- Mendoza-Cano, O., Danis-Lozano, R., Trujillo, X., Huerta, M., Ríos-Silva, M., Lugo-Radillo, A., Bricio-Barrios, J. A., Benites-Godínez, V., Cuevas-Arellano, H. B., Uribe-Ramos, J. M., Solano-Barajas, R., Cárdenas, Y., Venegas-Ramírez, J., Ríos-Bracamontes, E. F., García-Solórzano, L. A., Camacho-delaCruz, A. A., & Murillo-Zamora, E. (2025). Spatial patterns and clustering of dengue incidence in Mexico: Analysis of Moran's index across 2,471 municipalities from 2022 to 2024. *PLOS ONE*, 20(5), e0324754. <https://doi.org/10.1371/journal.pone.0324754>
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1), 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
- Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10), 1345–1359. <https://doi.org/10.1109/TKDE.2009.191>
- Raza, M. U., Kasamsumran, N., Ampol, R., Jariyapan, N., Iba, H., & Pora, W. (2026). Dilated YOLOv8 for blur-robust image classification on the MosquitoLarvae-7400 dataset. *IEEE Access*, 14, 37038–37054. <https://doi.org/10.1109/ACCESS.2026.3667960>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). IEEE. <https://doi.org/10.1109/CVPR.2016.91>
- Roboflow. (2024). *Roboflow platform for computer vision*. <https://roboflow.com>
- Sasmitha, H. I., Neoh, K.-B., Yusmalinar, S., Anggraeni, T., Chang, N.-T., Bong, L.-J., Putra, R. E., Sebayang, A., Silalahi, C. N., Tu, W.-C., & Ahmad, I. (2021). Ovitrap surveillance of dengue vector mosquitoes in Bandung City, West Java Province, Indonesia. *PLOS Neglected Tropical Diseases*, 15(10), e0009896. <https://doi.org/10.1371/journal.pntd.0009896>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, Article 60. <https://doi.org/10.1186/s40537-019-0197-0>
- Ultralytics. (2024). *Ultralytics YOLO documentation*. <https://docs.ultralytics.com>

| No. | Rol                                                            | Descripción                                                                                                                                                                                                      |
|-----|----------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|     |                                                                | <p><i>Nota 1: favor de eliminar los corchetes una vez colocado su nombre</i></p> <p><i>Nota 2: elimine el todo texto de apoyo que se proporcionó en los asteriscos de cada fila y debajo de cada nombre.</i></p> |
| 1   | Conceptualización<br>[Conceptualization]                       | Edson Raúl Campa Ramírez<br>Luis Amado González Vargas                                                                                                                                                           |
| 2   | Curación de datos<br>[Data curation]                           | Noé Alvarado Tovar<br>Silvia Adriana Vidaña Martínez                                                                                                                                                             |
| 3   | Análisis formal<br>[Formal analysis]                           | Edson Raúl Campa Ramírez<br>Rubén Hernández Rodríguez                                                                                                                                                            |
| 4   | Obtención de fondos<br>[Funding acquisition]                   | Noé Alvarado Tovar                                                                                                                                                                                               |
| 5   | Investigación<br>[Investigation]                               | Edson Raúl Campa Ramírez<br>Noé Alvarado Tovar                                                                                                                                                                   |
| 6   | Metodología<br>[Methodology]                                   | Edson Raúl Campa Ramírez<br>Luis Amado González Vargas<br>Rubén Hernández Rodríguez                                                                                                                              |
| 7   | Administración del proyecto<br>[Project administration]        | Noé Alvarado Tovar                                                                                                                                                                                               |
| 8   | Recursos<br>[Resources]                                        | Rubén Hernández Rodríguez<br>Silvia Adriana Vidaña Martínez                                                                                                                                                      |
| 9   | Software<br>[Software]                                         | Edson Raúl Campa Ramírez<br>Noé Alvarado Tovar                                                                                                                                                                   |
| 10  | Supervisión<br>[Supervision]                                   | Luis Amado González Vargas<br>Rubén Hernández Rodríguez                                                                                                                                                          |
| 11  | Validación<br>[Validation]                                     | Rubén Hernández Rodríguez<br>Silvia Adriana Vidaña Martínez                                                                                                                                                      |
| 12  | Visualización<br>[Visualization]                               | Edson Raúl Campa Ramírez<br>Noé Alvarado Tovar                                                                                                                                                                   |
| 13  | Redacción - borrador original<br>[Writing – original draft]    | Edson Raúl Campa Ramírez<br>Luis Amado González Vargas<br>Noé Alvarado Tovar                                                                                                                                     |
| 14  | Redacción - revisión y edición<br>[Writing – review & editing] | Noé Alvarado Tovar<br>Luis Amado González Vargas<br>Silvia Adriaa Vidaña Martinez                                                                                                                                |