

Sistema de etiquetado asistido para el entrenamiento de modelos de visión artificial usados en la detección de ninfas de mosca blanca

Juan Carlos Gonzalez Nava, Modesto Medina-Melendrez,
Diana Karina Jacobo Rubio

Instituto Tecnológico Nacional de México/Instituto Tecnológico de Culiacán,
México

[M19171353, D20171595, modesto.mm]@culiacan.tecnm.mx

Resumen. El monitoreo de la ninfa de mosca blanca (*Bemisia tabaci*) es esencial para prevenir pérdidas severas de cultivos; sin embargo, la identificación manual de ninfas sigue siendo prohibitivamente ineficiente. Si bien los modelos de visión computacional existentes exhiben una precisión robusta, carecen del nivel de detalle requerido para diferenciar entre instares biológicos o estados de vitalidad. Esta investigación introduce un sistema de etiquetado asistido por computadora diseñado a medida para la detección de la ninfa de mosca blanca, integrando una red neuronal de alta precisión para generar anotaciones automatizadas preliminares. La arquitectura del marco de trabajo comprende un interfaz de usuario que utiliza la partición de imágenes para mitigar la fatiga ocular, y una lógica del sistema diseñada para normalizar la escala física de las capturas de alta resolución. Este paradigma permite a los especialistas validar y refinar datos biológicos intrincados en lugar de ejecutar tareas manuales ex nihilo. La validez de esta propuesta se sustenta en la generación de un conjunto de datos estructurado. Al agilizar la curación de conjuntos de datos y minimizar la falibilidad humana, esta herramienta facilita la transición hacia la agricultura de precisión mediante intervenciones localizadas.

Palabras clave: Mosca blanca, agricultura, arquitecturas de aprendizaje profundo.

Assisted Labeling System for Training Computer Vision Models Used in Whitefly Nymph Detection

Abstract. Monitoring whitefly (*Bemisia tabaci*) nymphs is essential to prevent severe crop losses; however, manual identification of nymphs remains prohibitively inefficient. While existing computer vision models demonstrate robust accuracy, they lack the level of detail required to differentiate between biological instars or viability states. This research introduces a custom-designed, computer-assisted labeling system for whitefly nymph detection, integrating a high-precision neural network to generate preliminary automated annotations.

The framework's architecture comprises a user interface that employs image partitioning to mitigate eye strain, alongside system logic designed to normalize the physical scale of high-resolution captures. This paradigm enables specialists to validate and refine intricate biological data rather than performing manual tasks *ex nihilo*. The validity of this proposal is supported by the generation of a structured dataset. By streamlining dataset curation and minimizing human error, this tool facilitates the transition toward precision agriculture through localized interventions.

Keywords: Whitefly, agriculture, deep learning architectures.

1. Introducción

La mosca blanca (*Bemisia tabaci*) es una de las plagas más comunes y económicamente dañinas en la agricultura mundial. Su peligrosidad radica no solo en su movilidad y capacidad reproductiva en estado adulto, sino también que en su etapa inmadura (ninfa), afecta severamente la productividad de los cultivos mediante la succión de savia y la excreción de melaza, lo que debilita la planta y favorece la aparición de hongos [1].

Para superar la lentitud e imprecisión del monitoreo manual agrícola, es esencial automatizar la detección y así lograr una aplicación precisa de pesticidas. Esto requiere desarrollar herramientas de visión artificial capaces de procesar imágenes de alta resolución para identificar características morfológicas clave [2].

En este contexto, el valor de la IA reside en su capacidad para distinguir estados biológicos complejos, cuya eficacia depende directamente de la calidad de los datos [3]. Dado que el ruido o las etiquetas ambiguas empleadas para entrenar introducen sesgos que debilitan la capacidad de generalización del modelo de IA [4], la integridad del conjunto de datos no depende de la cantidad, sino de un preprocesamiento y etiquetado preciso.

El etiquetado de datos transforma imágenes de cultivos en información de entrenamiento mediante anotaciones semánticas, donde la precisión al identificar instares y la viabilidad (vivo/muerto) de la *Bemisia tabaci* (mosca blanca) es crítica para que la red neuronal aprenda patrones biológicos en lugar de ruido. A diferencia de otros ámbitos como la seguridad o la medicina, el análisis de microinsectos presenta una alta complejidad técnica debido al reducido tamaño y densidad de los especímenes, lo cual induce fatiga cognitiva y errores humanos [5]. En este contexto, resulta imperativo implementar sistemas de apoyo que optimicen el etiquetado, garantizando la creación de conjuntos de datos precisos para mejorar la eficiencia del monitoreo automatizado.

El etiquetado eficiente de microinsectos requiere segmentar las imágenes para optimizar detalles y reducir la fatiga visual [6]. Por esta razón, es necesario integrar una escala automática que asegure la precisión métrica y evolucionar hacia un proceso asistido por IA, donde el usuario valide el instar y el estado vital de la ninfa para garantizar registros de alta calidad. No obstante, se advierte que, aunque las plataformas actuales ofrecen una automatización avanzada y facilitan la creación de cuadros

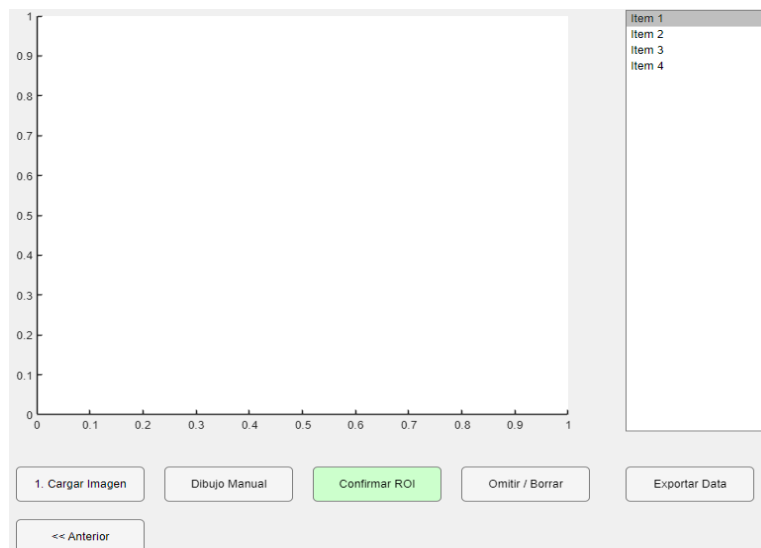


Fig. 1. Vista general de la interfaz gráfica desarrollada en MATLAB.

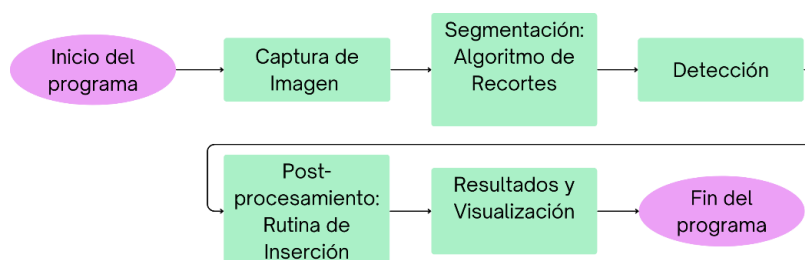


Fig. 2. Diagrama de flujo del sistema YOLO2[12].

delimitadores, estas presentan limitaciones críticas al aplicarse específicamente a la detección de este tipo de organismos [7].

A pesar de los avances en la detección de microinsectos, los detectores convencionales presentan limitaciones significativas ante imágenes degradadas por desenfoque o baja resolución, lo que compromete la precisión del etiquetado. Esta brecha tecnológica subraya la necesidad de implementar un sistema de apoyo para la generación de conjuntos de datos de alta fidelidad, asegurando la robustez necesaria para que las arquitecturas de aprendizaje profundo operen con eficacia en entornos agrícolas.

2. Trabajos previos

Han surgido diversas alternativas que implementan redes neuronales para automatizar la detección de plagas, destacando los trabajos de Laekeman et al. [8] el

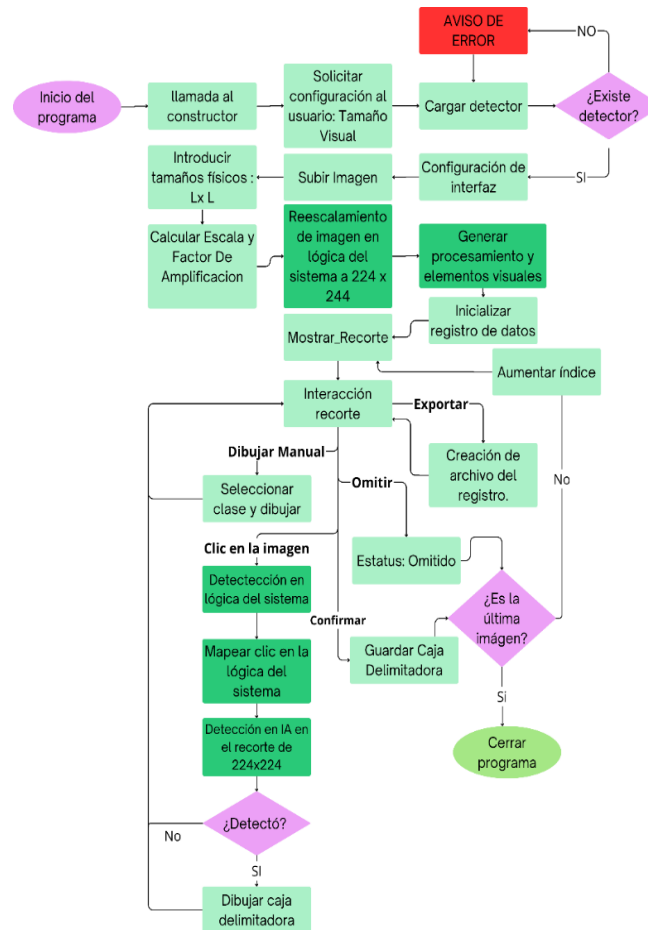


Fig. 3. Diagrama de flujo del sistema.

cual detecta la presencia de mosca blanca y de Huang et al. [9] donde se detectó mosca blanca, pulgones y trips. No obstante, estas soluciones suelen presentar precisiones asimétricas, situándose entre el 85% y 96%, o se limitan únicamente a detectar la etapa adulta de la plaga.

El trabajo de Jacobo [10] alcanzó un 97% de precisión en la detección de ninfas sobre hojas de tomate. Sin embargo, estos modelos presentan limitantes como, la incapacidad de diferenciar instares ninfales ni estados de vitalidad, factores clave dado que los primeros estadios son más susceptibles a los tratamientos, esto declarado por UC IPM [11].

Esta limitante surge de la falta de herramientas para que un experto interactúe con modelos de alta precisión y valide información masivamente. Superar este vacío es fundamental, pues la importancia de la IA radica en procesar volúmenes de datos con una velocidad y precisión inalcanzables para el monitoreo humano tradicional.

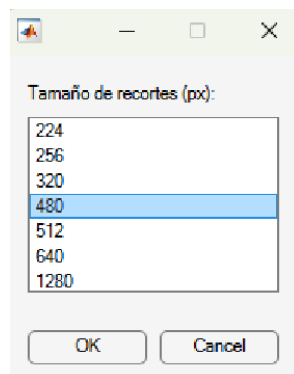


Fig. 4. Ventana de selección del tamaño de recortes con sus múltiples opciones.

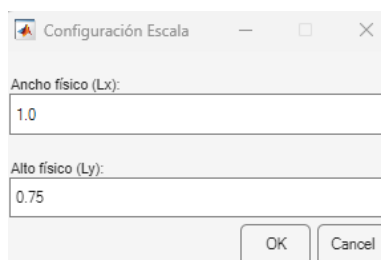


Fig. 5. Diálogo de interfaz gráfica de usuario (GUI).

3. Metodología

El sistema consiste en una interfaz gráfica interactiva diseñada para la creación, re-clasificación y validación de bases de datos mediante un flujo de trabajo semiautomatizado. El proceso comienza cuando el usuario carga una imagen con una definición mayor a las resoluciones válidas del sistema (240, 480, 512, 640, 1080) y define sus dimensiones físicas reales; con estos datos, el software calcula automáticamente un factor de amplificación para normalizar la escala de las imágenes y poderlas utilizar con un sistema detector de ninfas de mosca blanca previamente desarrollado, esta acción asegura que el modelo de visión artificial trabaje independientemente de la resolución de imagen utilizada.

Para agilizar el proceso de etiquetado, el sistema segmenta las imágenes de alta resolución en una cuadrícula de dimensiones procesables por la configuración propuesta por Jacobo [12], la cual está basada en YOLO. La integración de un modelo de clasificación asistida automatiza la generación de cajas delimitadoras al seleccionar un ejemplar, lo que reduce la carga de dibujo manual. Este proceso optimiza la validación de los datos al mantener la supervisión humana: el usuario conserva el control total para confirmar, corregir o descartar las detecciones propuestas por la IA.

La interfaz está constituida por un visor de alta precisión que adapta su tamaño según las necesidades del usuario y que está en función del sistema inteligente de destino, un



Fig. 6. Interfaz del etiquetador en MATLAB mostrando el primer recorte y la lista de registro.

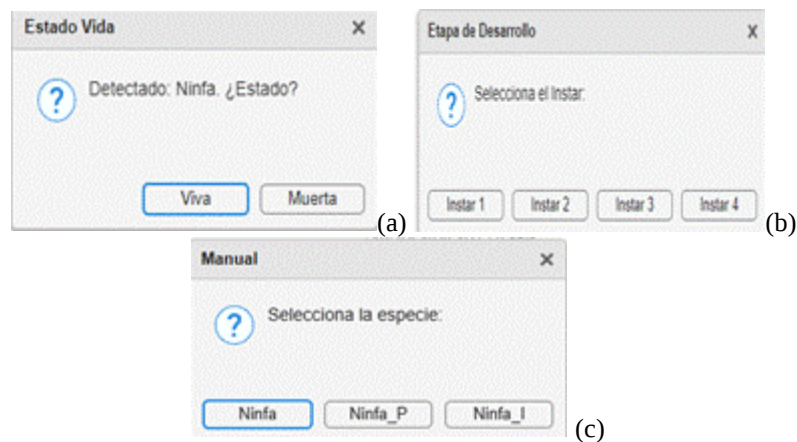


Fig. 7. Selección de clases de objetos. a) Selección de estado de vida; b) Selección de etapa de desarrollo; c) Selección de tipo.

panel de herramientas para la gestión de clases y una bitácora de seguimiento en tiempo real que organiza el estado de cada segmento de la imagen (pendiente, confirmado u omitido).

Por último, el sistema genera datos de entrenamiento validados tras el cierre de sesión, operando bajo una estructura dual de Interfaz de Usuario y Lógica de Datos.



Fig. 8. Caja delimitadora de color cian que delimita al insecto.



Fig. 9. Caja delimitadora mal posicionada.



Fig. 10. Corrección de la caja delimitadora por medio de los puntos de ajuste.

3.1. Interfaz de usuario

La interfaz de usuario fue desarrollada en MATLAB. Esta consiste en un menú de control intuitivo que prioriza la eficiencia operativa en la generación de conjuntos de datos. El diseño se organiza en un visor interactivo izquierdo de tamaño dinámico para la visualización a escala real y detección asistida, complementado por un panel de registro a la derecha que permite monitorear el progreso del proceso mediante los estados PENDIENTE, CONFIRMADO y OMITIDO.

El sistema integra una barra de herramientas centralizada que coordina el flujo de trabajo completo, abarcando desde la carga de imágenes y el ajuste de escalas hasta el dibujo y la exportación al espacio de trabajo.

Al mismo tiempo, la barra de estado comunica cualquier avance mediante retroalimentación en tiempo real. Esta arquitectura transforma la inspección compleja de imágenes de alta resolución en un proceso semiautomático y ágil, permitiendo al usuario validar detecciones, ajustar parámetros de clasificación y navegar secuencialmente por los segmentos con total coherencia técnica, esto se ilustra en la Figura 1.

```

1: [OMITIDO] 1 objs
2: [OMITIDO] 0 objs
3: [PENDIENTE] 0 objs
4: [PENDIENTE] 0 objs
5: [PENDIENTE] 0 objs
6: [PENDIENTE] 0 objs
7: [PENDIENTE] 0 objs
8: [PENDIENTE] 0 objs
9: [PENDIENTE] 0 objs
    
```

Fig. 11. Actualización del estado CONFIRMADO tras el proceso de guardado.



Fig. 12. Caja delimitadora dibujada sobre un ejemplar no detectado automáticamente.

3.2. Lógica del sistema

La lógica del sistema se desarrolló bajo el paradigma de Programación Orientada a Objetos (POO) en MATLAB. Su función principal consiste en la normalización de imágenes de alta resolución a partir de parámetros físicos reales en pulgadas (Lx y Ly) y la gestión de la lógica de procesamiento.

La arquitectura implementa un algoritmo de segmentación que divide la imagen en recortes técnicos de 224 x 224 píxeles, dimensiones optimizadas para su procesamiento mediante la arquitectura de aprendizaje profundo propuesta por Jacobo [12] (el diagrama de flujo de dicho desarrollo se muestra en la Figura 2), la cual utiliza un modelo YOLO V2 soportado por una red ResNet-50 que extrae sus características desde la capa relu_40, un punto de activación no lineal encargado de filtrar y procesar los patrones visuales para optimizar la identificación de la plaga.

Como parte del entrenamiento, la red fue capacitada para detectar ninfas de mosca blanca, proveyendo tres clases posibles: Ninfa, Ninfa_Posible o Ninfa_Incompleta. Tomando esta configuración como base, el sistema propuesto extiende su funcionalidad al integrar un entorno de gestión de datos, donde la segmentación y el mapeo de coordenadas se automatizan para garantizar la trazabilidad de cada resultado. Este proceso permite una clasificación precisa de categorías biológicas a través del mapeo de coordenadas entre la interacción del usuario en la interfaz y el procesamiento en la lógica del sistema, lo cual establece las bases para la evaluación de desempeño.

Finalmente, los resultados se conforman y estructuran en una tabla de registro. Este sistema de almacenamiento asegura la persistencia de los detalles técnicos de cada zona de interés, permitiendo que el usuario exporte su conjunto de datos final de manera ágil

ID_Recorte	Ruta	ROI				Tipo	Estado	Instar
		1	2	3	4			
1	"C:\Detector\02.bmp"	226.9765	307.9681	64.9933	59.9938	"Ninfa_1"	"Viva"	"Instar 2"
1	"C:\Detector\02.bmp"	75.9921	227.9764	43.9954	68.9928	"Ninfa"	"Muerta"	"Instar 3"
2	"C:\Detector\02.bmp"	172.9821	288.9202	82.7257	56.9941	"Ninfa_1"	"Muerta"	"Instar 2"
2	"C:\Detector\02.bmp"	227.9764	121.9376	58.9939	46.9951	"Ninfa"	"Viva"	"Instar 3"
3	"C:\Detector\02.bmp"	298.9690	295.8697	35.9963	28.9970	"Ninfa"	"Muerta"	"Instar 3"
3	"C:\Detector\02.bmp"	277.9712	336.8655	32.9966	22.9976	"Ninfa"	"Muerta"	"Instar 2"
3	"C:\Detector\02.bmp"	370.9615	436.8551	36.9962	25.9973	"Ninfa"	"Viva"	"Instar 4"
4	"C:\Detector\02.bmp"	334.9653	428.8062	27.9971	30.9968	"Ninfa"	"Muerta"	"Instar 2"
4	"C:\Detector\02.bmp"	331.9656	340.8153	24.9974	34.9964	"Ninfa"	"Viva"	"Instar 4"
4	"C:\Detector\02.bmp"	279.9710	301.8193	27.9971	28.9970	"Ninfa"	"Viva"	"Instar 3"

Fig. 13. Estructura de la tabla de datos exportada en MATLAB.

y simplificada. En el diagrama de la Figura 3 se detalla la secuencia lógica de la carga de imágenes, su segmentación y la actualización de la tabla de registros.

4. Funcionamiento del sistema

Mediante la interfaz en MATLAB, el usuario define las dimensiones de los recortes y la resolución de almacenamiento. Esta parametrización asegura la versatilidad del conjunto de datos generado, garantizando su plena compatibilidad con diversas arquitecturas de redes neuronales, como YOLO (Figura 4).

Tras elegir el tamaño, se accede a la interfaz de carga de imagen. El sistema solicita las dimensiones reales (Lx, Ly) en pulgadas (Figura 5) para normalizar la escala espacial; esto asegura que el detector de IA procese los objetos con el tamaño en píxeles exacto de su entrenamiento, independientemente de la resolución o distancia de la toma. Finalmente, se carga el archivo y se inicializa la lista de registros (Figura 6).

Con la imagen cargada, el sistema permite la interacción directa: al hacer clic, se ejecuta el detector automático para identificar las ninfas, permitiendo su clasificación inmediata mediante etiquetas específicas. Al detectar una ninfa, este le pedirá al usuario que asigne su clasificación correcta de estado, instar y tipo, esto se observa en la Figura 7. Esta estructura de etiquetado ha sido diseñada para asegurar que la información recolectada sea precisa, ya que, al capturar estas variables, el usuario puede transformar los datos brutos en acciones concretas de intervención (como la selección del tratamiento y el momento óptimo de aplicación), basándose en el ciclo de vida y la especie identificada en cada etapa del desarrollo.

Tras la detección como se observa en la Figura 8, el sistema permite validar y refinar el resultado antes de su almacenamiento definitivo en el caso que la caja delimitadora no sea el adecuado véase en la Figura 9. El usuario tiene la posibilidad de editar la caja delimitadora manualmente mediante los puntos de ajuste (nodos en el recuadro cian) que puede visualizarse los cuales se miran en la Figura 10 para garantizar una

delimitación precisa del espécimen. Una vez que se guarda el resultado, el estado del recorte cambia automáticamente a "CONFIRMADO" esto se puede observar en la Figura 11, integrándose formalmente al conjunto de datos como un dato verificado.

Como se muestra en la Figura 12, el sistema permite la intervención manual cuando ocurren casos donde el detector no logra identificar automáticamente una ninfa, El usuario puede dibujar la caja delimitadora e ingresar las clasificaciones correspondientes, garantizando así la integridad de los datos.

5. Resultados

Como se ilustra en la Figura 13, los resultados forman una tabla de MATLAB donde cada fila representa una detección individual, asegurando la trazabilidad mediante el ID_Recorte y la ruta de origen. La localización espacial de cada espécimen se define mediante una caja delimitadora o región de interés (ROI por sus siglas en inglés Region of Interest) estructurada en cuatro parámetros fundamentales: las columnas 1 y 2 representan las coordenadas de origen (x, y) del vértice superior izquierdo de la detección, mientras que las columnas 3 y 4 especifican el ancho y el alto de la caja delimitadora en píxeles. Estos valores se vinculan de forma estructurada con tres dimensiones de metadatos biológicos: tipo, condición vital (Estado) y etapa de desarrollo (Instar). Esta organización no solo simplifica el análisis estadístico poblacional, sino que constituye un conjunto de datos etiquetados optimizado para el entrenamiento de modelos de visión artificial, permitiendo realizar filtrados automáticos basados en criterios taxonómicos o fenológicos específicos.

El proceso de generación del conjunto de datos destacó por su rapidez, gracias a la alta tasa de aciertos del algoritmo de detección. Se observó que la principal dificultad radicó en el reconocimiento de ninfas con baja nitidez. No obstante, este inconveniente no compromete la integridad del producto final. El conjunto de datos queda listo para ser sometido a pruebas posteriores de rendimiento en detectores inteligentes de ninfas de mosca blanca existentes.

6. Conclusiones

Este sistema simplifica el etiquetado de imágenes mediante un asistente semiautomático de generación de trazos. En este sentido, el usuario solo debe validar o ajustar las cajas delimitadoras y las etiquetas de los objetos detectados o en su defecto generar una propuesta nueva. Asimismo, la herramienta incrementa la eficiencia operativa en la creación de conjuntos de datos para sistemas inteligentes. Debido a su diseño versátil, es posible procesar múltiples ninfas independientemente de sus tamaños. Por consiguiente, se garantiza la obtención de conjuntos de datos para el entrenamiento de modelos de visión artificial. Por lo que se puede confirmar que, la automatización de estos procesos permite simplificar el flujo de trabajo técnico, priorizando así las etapas críticas de la investigación.

Referencias

1. Oliveira, M.R.V., Henneberry, T.J., Anderson, P.: History, Current Status, and Collaborative Research Projects for *Bemisia tabaci*. *Crop Prot.*, 20(9), pp. 709–723 (2001). Doi: 10.1016/S0261-2194(01)00108-9.
2. Huerta-Mora, E.A., Tostado-Ramirez, M.I., Estrada Lizárraga, R.: Sistema de detección de enfermedades en plantas para la reducción de uso de pesticidas y plaguicidas en la región de Sinaloa. *Revista Ciencias de la ingeniería*, 1(1), pp. 06–12 (2025)
3. Adelusola, M.: Data Quality Over Model Complexity: Rethinking the AI Performance Paradigm (2025)
4. Nogueira, K.: Data-Centric Benchmark for Label Noise Estimation and Ranking in Remote Sensing Image Segmentation. *arXiv preprint* (2026)
5. Monarch, R.: *Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI*. Manning Publications (2021)
6. Venkatesh, S., Robbins, J., Scott, A.D.: Evaluation of Digital Pathology Workflows: From Slides to Screens. *Journal of Pathology Informatics*, 8(1), pp. 24 (2017)
7. CVAT.ai: Best Open-Source Image Annotation Tools (2024)
8. Laekeman, B., Bonte, J., Dermauw, W.: Species-Level Detection of Thrips and Whiteflies on Yellow Sticky Traps Using YOLO-Based Deep Learning Detection Models. *Front. Plant Sci.*, 16 (2025). Doi: 10.3389/fpls.2025.1668795.
9. Huang, Y., Liu, Z., Zhao, H.: YOLO-YSTS: An Improved YOLOv10n-Based Method for Real-Time Field Pest Detection. *Agronomy*, 15(3) (2025). Doi: 10.3390/agronomy15030575.
10. Jacobo-Rubio, D.K., Medina-Melendrez, M.: Sistema inteligente para la detección de ninfas de mosca blanca presentes en hojas de plantas. *Research in Computing Science*, 152(9), pp. 187–200 (2023)
11. University of California Integrated Pest Management Program: UC IPM Statewide Integrated Pest Management Program (2026)
12. Jacobo Rubio, D.K.: Sistema inteligente para la detección y conteo de ninfas de mosca blanca presentes en hojas de plantas. Tesis de Maestría, Inst. Tecnol. de Culiacán (2023)