

Sistema inteligente para la detección de ninfas de mosca blanca presentes en hojas de plantas

Diana Karina Jacobo-Rubio, Modesto Medina-Melendrez

Tecnológico Nacional de México,
Instituto Tecnológico de Culiacán,
México

{M20171595, modesto.mm}@culiacan.tecnm.mx

Resumen. La FAO revela en sus estadísticas que 2570 millones de persona dependen de la agricultura para su subsistencia. Un factor de riesgo para esta son las plagas, generando pérdidas de hasta un 40%. Una de las plagas más comunes en regiones tropicales y subtropicales es la mosca blanca *Bemisia tabaci* (Gennadius), que afecta a más de 600 especies de plantas. Para tener un control de esta plaga es necesario realizar su monitoreo, el cual es recomendable realizarlo en su etapa ninfal ya que en esta etapa se pueden obtener estimaciones precisas del nivel de infestación. Este artículo propone un monitoreo automático entrenando y usando un sistema inteligente para detectar ninfas de mosca blanca en hojas de plantas utilizando la red YOLO V2. El sistema se validó utilizando un banco de imágenes de hojas de tomate infestadas con mosca blanca, y se logró una precisión promedio de hasta 97% al detectar esta plaga.

Palabras clave: Plagas, mosca blanca, detección y sistemas inteligentes.

Intelligent System for the Detection of Whitefly Nymphs Presents on Plant Leaves

Abstract. FAO reveals that 2.57 billion people subsist on agriculture. A risk factor for it are pests, generating losses of up to 40%. One of the most common pests in tropical and subtropical regions is the whitefly *Bemisia tabaci* (Gennadius), affecting more than 600 plant species. To control this pest, it can be monitored in nymphal stage, since it estimates the level of infestation at this stage. This paper proposes an automatic monitoring by training and using an intelligent system for detecting whitefly nymphs present on leaves using the YOLO V2 network. After systems's validation using an image bank of infested tomato leaves, an average precision of 97% in detection was obtained.

Keywords: Pests, whitefly, detection and intelligent system.

1. Introducción

La Organización de las Naciones Unidas para la Alimentación y la Agricultura (FAO) revela en sus estadísticas que, a principios del nuevo milenio, 2570 millones de personas dependen de la agricultura, caza, entre otras actividades para su subsistencia.

Estas personas representan el 42% de la humanidad [1]. Un factor de riesgo para la agricultura son las plagas, estas pueden generar pérdidas de hasta un 40 por ciento de acuerdo con la FAO [2]. Una de las plagas mayormente distribuidas en regiones tropicales y subtropicales del mundo es la mosca blanca *Bemisia tabaci* (Gennadius), esta afecta a más de 600 especies de plantas cultivadas y silvestres [3].

En la última década la mosca blanca ha causado millones de dólares en pérdidas de cultivos en agroecosistemas a lo ancho del mundo [4]. Existen distintos métodos tanto químicos como biológicos para controlar las plagas y para poder aplicarlos es necesario mantener un monitoreo constante, en el caso de la mosca blanca es recomendable realizarlo en su etapa ninfal, ya que en la etapa adulta estas no suelen quedarse en las hojas y esto provoca estimaciones poco precisas del nivel de infestación [5].

El método más utilizado para monitorear poblaciones de plagas en estudios de salud de cultivos es mediante una inspección visual realizada por trabajadores. Este método resulta ser complicado, requiere mucho tiempo, suele ser extenuante para los trabajadores y por ello es propenso a errores humanos [6]. Por lo que es necesario desarrollar un instrumento automático para monitorear eficientemente las plagas en siembras e invernaderos.

La agricultura 4.0 busca integrar un conjunto de tecnologías, dispositivos, protocolos y paradigmas computacionales para mejorar los procesos agrícolas. En este nuevo enfoque se utilizan técnicas de procesamiento digital de imágenes como una herramienta que permite identificar de manera temprana las plagas o enfermedades en los cultivos [7].

Tanto los agricultores como las empresas buscan aumentar la producción y reducir los residuos, como resultado, la IA está emergiendo constantemente como parte de la evolución tecnológica de la industria agrícola [8].

El uso de técnicas de procesamiento de imágenes (PDI) e inteligencia artificial (IA) es una posibilidad para desarrollar sistemas de monitoreo automático. Para implementar este sistema es necesario contar con cámaras para capturar imágenes de las plantas o en su defecto de las hojas que son clave para detectar las ninfas de mosca blanca, luego es necesario aplicar técnicas de procesamiento de imágenes para poder detectar las ninfas.

2. Trabajos relacionados

En la literatura existen distintos sistemas desarrollados para la detección de mosca blanca adulta, pero son pocos los sistemas reportados para la detección de moscas blancas en su estado ninfal. Uno de los trabajos reportados se describe en el artículo escrito por Lino [9], el cuál utiliza técnicas de detección de patrones mediante algoritmos Haar-like Features (Haar) y Local binary patterns (LBP) para detectar y contar ninfas de mosca blanca en frijol ejotero.

En este artículo, Lino reporta resultados con un alto índice de falsos positivos provocados por la misma morfología de la planta, lo que afecta al conteo real de esta plaga. Una alternativa diferente fue reportada por Bernal [10], el cual propone un sistema de captura, detección de formas elípticas irregulares y técnicas de combinatoria para la detección y conteo de mosca blanca en estado ninfal sobre hojas de tomate. Este sistema detecta y cuenta las ninfas de mosca blanca sobre hojas de tomate de forma automática, sin embargo, este sistema cuenta con una precisión menor al 85%.

Tabla 1. Técnicas utilizadas para detectar plagas.

Técnicas utilizadas	Precisión
Red neuronal convolucional	92.4%
Codificación multitarea escasa, técnica de aprendizaje multinúcleo	Color:70.2%, textura:63.5, forma:80.2%
Clasificador multicapa basada en el clasificador de umbral aleatorio	Rango de reconocimiento: 85%
Aprendizaje de múltiples instancias	59.8%
Aprendizaje profundo con imágenes UAV	91.2%
Método de aprendizaje residual profundo	Clasificación de 98.6%
Clasificación con Máquinas vectoriales de soporte (SMV)	Error menor al 2.5%
Cascada Haar	83%
Red neuronal convolucional por regiones más rápida (RCNN)	98%

La aplicación de tecnologías e inteligencia artificial (IA) en la agricultura se ha desarrollado a lo largo de los años como parte de las soluciones hacia la mejora de la productividad agrícola y así poder satisfacer la enorme demanda alimentaria que aumenta año con año. Estas tecnologías e implementación de la IA no solo permiten a los agricultores mejorar la eficiencia, sino que también mejoran la calidad, la cantidad y garantizan una comercialización más rápida de los cultivos [11].

En la tabla 1 se muestra una recopilación de técnicas para detectar plagas mediante sistemas inteligentes realizada por Gómez [12] y Liakos [13]. De acuerdo con el estudio de los antecedentes, existen distintos trabajos relacionados con la detección de mosca blanca en cultivos, siendo pocos los sistemas relacionados con la detección de la mosca blanca en estado ninfal.

Entre los sistemas que detectan mosca blanca en su estado ninfal se encuentra el sistema propuesto por Bernal, este logra una precisión menor al 85%. En los sistemas desarrollados para la detección de mosca blanca en estado ninfal también se encuentra el sistema propuesto por Lino, el cual detecta esta plaga en hojas de frijol ejotero, el inconveniente es que presenta altos índices de falsos positivos como lo indica en su artículo [9].

En el caso de los sistemas que utilizan técnicas de IA y/o PDI, estos son capaces de detectar eficientemente plagas, pero aún no se ha desarrollado un sistema que utilice estas técnicas en conjunto para detectar moscas blancas en estado ninfal sobre hojas de plantas. Por ello, en este artículo se describe un sistema inteligente para la detección de ninfas de mosca blanca presentes en hojas de plantas cuyo propósito es mejorar la precisión respecto a los trabajos previamente reportados en la literatura.

3. Metodología propuesta

Como se puede observar en la tabla 1, existen distintas técnicas que pueden lograr una alta precisión para detectar plagas, una de las que mayor precisión presenta es la técnica RCNN. La red llamada You Only Look Once versión 2 (YOLO V2) es del tipo

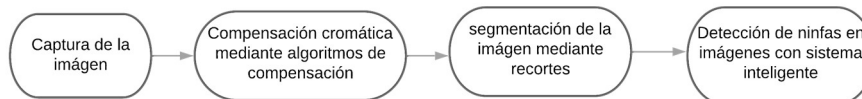


Fig. 1. Diagrama de flujo del sistema inteligente para la detección de mosca blanca en estado ninfal.

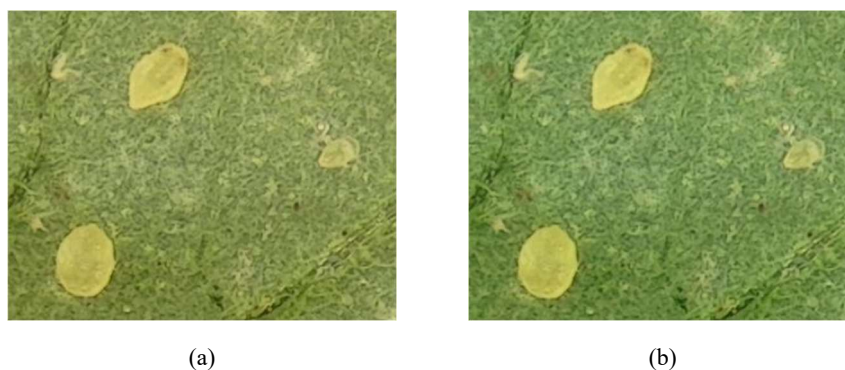


Fig. 2. Compensación cromática. (a) Fragmento de una imagen original (b) Fragmento de una imagen compensada.

RCNN y se utiliza en el desarrollo de este proyecto, ya que esta se puede configurar, generar y entrenar solamente modificando ciertos parámetros de entrenamiento.

Existen distintas plataformas de programación como MATLAB y Python que cuentan con herramientas para configurar una red YOLO V2. Para crear una red de detección de objetos YOLO V2 es necesario especificar los parámetros del tamaño de imagen de entrada, número de clases, número de cajas de anclaje, número de épocas, tamaño del mini lote de entrenamiento y las épocas.

Primero, para especificar el tamaño de imagen de entrada el cuál es de 224x224x3 pixeles, se consideró el tamaño mínimo requerido de la red, ya que al momento de entrenar la red el coste computacional se ve reducido al procesar los datos. Conforme se avance en esta sección se estará brindando más información acerca de la configuración de los otros parámetros.

El sistema desarrollado está compuesto por cuatro módulos, la Fig. 1 representa el diagrama con las etapas de procesamiento que se consideran importantes para poder realizar la detección de ninfas de mosca blanca presentes en hojas de plantas. Las siguientes secciones están divididas en correspondencia a los módulos del sistema propuesto, por lo que en cada sección se detalla que es lo que se necesita para generar cada módulo, qué es lo que hace y el resultado de este.

3.1. Captura de imagen

En este módulo se implementa un sistema de captura de imágenes constituido por un sistema óptico de reducción, ya que la captura total de la escena en la imagen debe

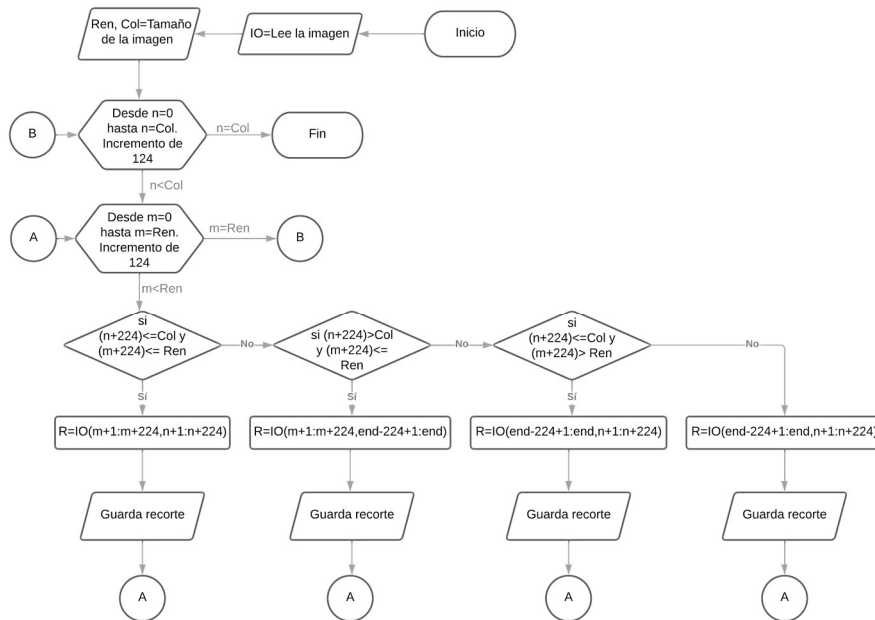


Fig. 3. Algoritmo utilizado para hacer recortes.

abarcar un área de 1 in^2 ; una cámara; un sistema de iluminación y una base para poder capturar las muestras.

El sistema óptico de reducción y captura de imágenes está compuesto por una cámara con puerto USB para captura y transmisión de imágenes al ordenador y lentes de reducción para ajustar una escena relativamente grande al área sensible del sensor de la cámara. Para poder obtener un mejor enfoque en las capturas de imágenes. La cámara cuenta con una apertura para limitar la cantidad de luz que entra al lente.

En cuanto al sistema de iluminación se utilizó un anillo de leds que cumple con la iluminación estándar D65, ya que es la requerida por la paleta de colores de referencia. Las características técnicas de la cámara y el sistema de lentes utilizan un sensor True color CMOS, distancia entre pixeles de $2.2 \text{ um} \times 2.2 \text{ um}$, una resolución de 2592 columnas x 1944 renglones y un aumento de $0.12\times - 2\times$. Se utiliza un ordenador con el software de MATLAB versión R2019a instalado.

Las especificaciones de la computadora son las siguientes: el modelo del ordenador es una Torre Dell Precision 3620, el procesador es Intel (R) Core (TM) i7-7700 CPU @ 3.60GHz, la RAM instalada es de 16.0 GB, utiliza un sistema operativo de 64 bits, y sistema operativo Windows 10 de 64 bits.

3.2. Compensación cromática

La compensación cromática estandariza las medidas de color y así poder caracterizar correctamente la croma del objeto de estudio (las ninfas de mosca blanca). Las imágenes capturadas dependen de diferentes variables de la cámara, ya sea por el tipo

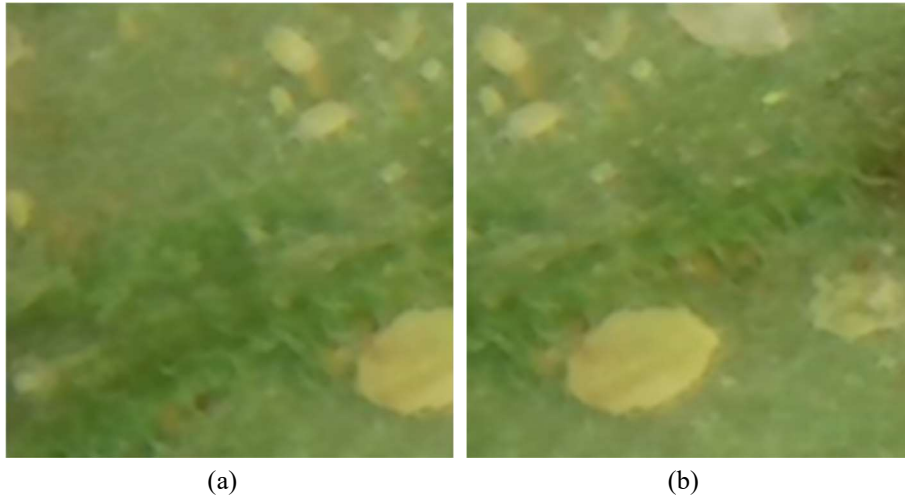


Fig. 4. Imágenes del banco de imágenes. (a) ninfa recortada (b) ninfa completa.

de sensor que utiliza (CMOS o CCD), la posición de los fotosensores, el patrón de Bayer qué utilicen, entre otros factores.

En este caso se realizó la compensación cromática mediante los polinomios caracterizados por Bernal [10], el cual utilizó como referencia la paleta de colores Macbeth Colorcheck para parametrizar el polinomio de la Ecuación 1 que se utiliza para compensar las imágenes capturadas:

$$H_r = -506.3H_m_8 + 2060.7H_m_7 - 3407H_m_6 + 2926.4H_m_5 - 1383.5H_m_4 + 350.7H_m_3 - 43.4H_m_2 + 3.357H_m - 0.0373, \quad (1)$$

donde H_r es el valor de H (Hue) resultante y H_m es el valor medido de H. Un ejemplo de aplicar la compensación cromática se muestra en la Fig. 2, en la cual se observa la imagen original en el inciso (a) y la imagen compensada cromáticamente en el inciso (b).

3.3. Segmentación de la imagen

Algoritmo de recortes

El primer paso para entrenar una red es generar una base de datos. La base de datos necesita un banco de imágenes, por lo que posterior a la captura de imágenes de 2592×1944 píxeles se realizaron recortes de $224 \times 224 \times 3$ de ellas para conformar el banco de imágenes.

Los recortes se realizaron con traslapes de 100 renglones y 100 columnas en sus extremos, esto para poder contener versiones completas de todas las ninfas, incluyendo las ninfas más grandes que tienen un diámetro máximo de hasta 99 píxeles. En la Fig. 3 se puede observar el diagrama de flujo del algoritmo utilizado para segmentar la imagen original.



Fig. 5. Ejemplo de etiquetado de objetos a detectar.

Se constituyó un banco de imágenes de 2,016 imágenes de $224 \times 224 \times 3$ píxeles. La Fig. 4 representa un ejemplo de recortes de imágenes, donde podemos ver a la misma ninfa en dos recortes vecinos, en el inciso (a) se observa a una ninfa cortada y en el inciso (b) se observa la ninfa completa.

3.4. Detección de ninfas con sistema inteligente

Para poder desarrollar el sistema inteligente capaz de detectar las ninfas de mosca blanca fue necesario crear un banco de imágenes de hojas infestadas y un banco de entrenamiento, configurar la red neuronal a utilizar, para posteriormente realizar su entrenamiento. Finalmente, se utiliza la red entrenada para lograr la detección de ninfas de mosca blanca.

Creación de base de datos

Se utilizó una interfaz amigable para cargar el banco de imágenes y etiquetar a las ninfas, un ejemplo de imagen cargada y objetos etiquetados se muestra en la Fig. 5. Se definieron 3 tipos de objetos a detectar y sus etiquetas correspondientes. Las etiquetas se definen por los tres casos morfológicos de las candidatas a ser ninfas que se presentan en las imágenes de hojas infestadas, y estas son: “Ninfa”, “Ninfa_I” y “Ninfa_P”.

La etiqueta “Ninfa” representa a aquellas que tienen la característica de la forma y color de una ninfa, la cuál debe ser ovalada y de un color más amarillo que el de la hoja; la etiqueta “Ninfa_I” representa a las ninfas que tienen las mismas características que la etiqueta “Ninfa”, pero esta se encuentra en los bordes de la imagen, por ello fue recortada; y la etiqueta de “Ninfa_P” representa a las candidatas a ser ninfas que cuenten con solo una de las característica deseadas, ya sea el color o la forma de la ninfa.

Creación de banco de entrenamiento

Para crear el banco de entrenamiento se necesita declarar la etiqueta, cargar el banco de imágenes, seleccionar la etiqueta que se quiere utilizar, y encerrar el objeto a etiquetar en un cuadro delimitador de región. Cuando se tienen todas las imágenes

	1	2	3	4
	imageFilename	Ninfa	Ninfa_I	Ninfa_P
1	'C:\Users\dikaj\D...	[188,48,34,46;62,92,26,20]	[163,155,6...	[]
2	'C:\Users\dikaj\D...	3x4 double	[]	[123,205,2...
3	'C:\Users\dikaj\D...	[128,191,30,22]	[164,32,61,...	[]
4	'C:\Users\dikaj\D...	3x4 double	[]	3x4 double
5	'C:\Users\dikaj\D...	[85,126,60,74;127,66,31,23]	[]	[82,111,15,...
6	'C:\Users\dikaj\D...	[110,116,56,84;75,47,31,21]	[]	[184,13,21,...
7	'C:\Users\dikaj\D...	[111,81,32,21]	[1,117,38,8...	[59.505527...
8	'C:\Users\dikaj\D...	[85,2,59,72]	[]	[]
9	'C:\Users\dikaj\D...	[173,190,21,34]	[109,1,56,7...	[]
10	'C:\Users\dikaj\D...	[125,81,56,81;49,191,22,33]	[1,1,39,75]	[]
11	'C:\Users\dikaj\D...	[1,81,57,82;129,51,30,26]	[205,211,2...	4x4 double
12	'C:\Users\dikaj\D...	[49,67,22,33;140,114,37,21]	[102,193,7...	3x4 double

Fig. 6. Banco de entrenamiento.

etiquetadas se genera la base de datos formando una tabla como se muestra en la Fig. 6, de la cual se extraen los diferentes elementos del banco de entrenamiento.

En la tabla de la Fig. 6, la primera columna representa la dirección de la imagen del banco de imágenes, en la segunda, tercera y cuarta columna se guardan los valores de las cajas delimitadoras de región de los objetos que se etiquetaron y que corresponden a las diferentes clases de objetos definidas (columna 2-Ninfa, columna 3-Ninfa_I y columna 4 Ninfa_P).

Cada caja delimitadora guarda los valores de su posición o coordenada (x, y) en la imagen, su ancho y su alto. Al utilizar la herramienta de etiquetado no se visualiza el tamaño de la caja de región de interés, lo que impidió discernir entre lo huevecillos y las ninfas de mosca blanca. Por lo que se etiquetaron todos los objetos que cumplen con forma y color.

Posterior al etiquetado, se utilizó un algoritmo de descarte de candidatas a ser ninfas si es que las cajas delimitadoras de región no cumplían con el tamaño mínimo requerido para contener las ninfas más pequeñas a ser detectadas.

Lo que se realizó en este caso fue un recorrido en la tabla de entrenamiento (Fig. 6) verificando el área de las cajas delimitadoras de región de interés (ancho por alto) y si estas resultaban más pequeñas que el área de la ninfa más pequeña (210 pixeles), estas candidatas fueron eliminadas. La lógica del algoritmo se muestra en la Fig. 7.

La Fig. 7 es un diagrama que representa la lógica del algoritmo de descarte de candidatas a ser ninfas, donde Ninfa representa la columna en la tabla de entrenamiento (Fig. 6) que contiene a las diferentes cajas delimitadoras que encierran a las ninfas y Ninfa_P representa a la columna en la tabla de entrenamiento que contiene a las diferentes cajas delimitadoras que encierran a las posibles ninfas y Am representa el área menor de la ninfa más pequeña.

Nn es el índice del ciclo que realiza el recorrido de los renglones en la tabla de entrenamiento y k es el índice que se utiliza para acceder a cada caja delimitadora contenida en la celda dentro del renglón Nn en la tabla de entrenamiento.

A cada caja delimitadora se le calcula el área, y se almacena en AN para la clase de objeto ninfa y en AN_P para la clase de objeto posibles ninfas. Posteriormente, estos

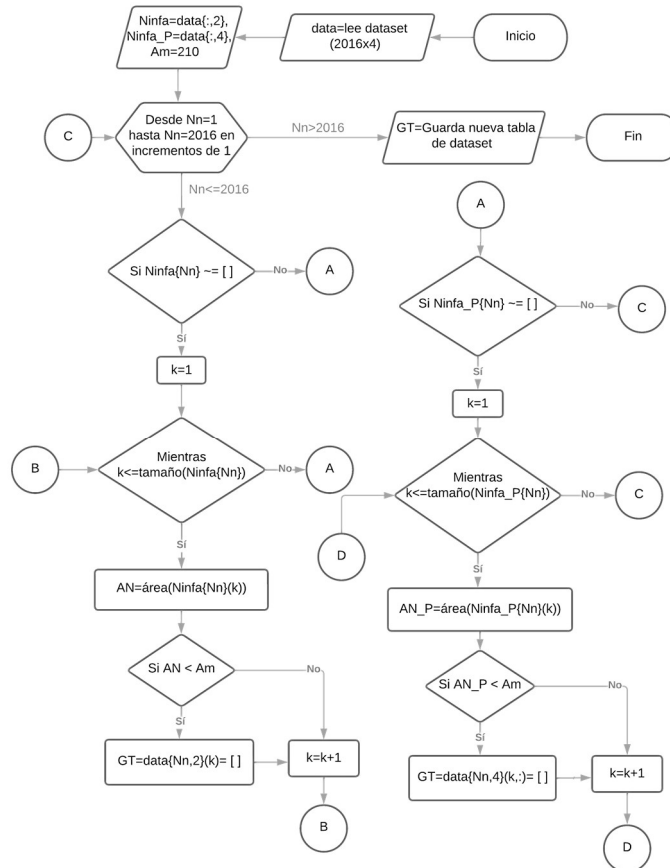


Fig. 7. Algoritmo de descarte de candidatas a ser ninfas.

valores se comparan con A_m y en caso de ser menores a este, se eliminan las cajas delimitadoras a las que corresponden o en el caso contrario se mantienen en la tabla de entrenamiento nueva.

Parámetros de configuración

Los parámetros de configuración de la red YOLO V2 son las cajas de anclaje, las épocas a utilizar y el tamaño del mini lote de entrenamiento. Las entradas del sistema son el tamaño de las imágenes de entrada, el banco de entrenamiento, el número de etiquetas o clases de objetos, un modelo preentrenado como capa de extracción y una capa de extracción de características.

El tamaño de la imagen de entrada es de $224 \times 224 \times 3$ como el tamaño de los recortes de imagen, el banco de entrenamiento que corresponde a la tabla de la Fig. 6, el número de etiquetas o clases de objetos corresponde a las que se muestran en la Fig. 3.

En el caso de la capa de extracción se seleccionó la resnet50, ya que es una red neuronal residual que ha demostrado ser útil para aumentar el rendimiento de las redes



Fig. 8. Resultado del detector operando sobre una imagen del banco de imágenes.

neuronales con propósito de visión artificial y reconocimiento de objetos al realizar saltos entre capas.

Por último, para la capa de extracción de características se seleccionó una relu 40, esta genera mapas de características que se reducen en un factor de 16, la cual es una buena compensación entre la resolución espacial y la fuerza de las características extraídas. Un parámetro que se puede modificar es la cantidad de cajas de anclaje, este parámetro se utiliza para generar las cajas de anclaje con el tamaño adecuado que se posicionarán en cada celda de la imagen.

Teniendo los datos de entrada se definen las opciones de entrenamiento para la red neuronal, estas opciones se pueden modificar para generar diferentes configuraciones, en este proyecto lo que se modificó fueron las épocas y el mini lote de entrenamiento, ya que es recomendable que los otros parámetros de la configuración para la red YOLO V2 se mantengan es sus valores originales.

Una época representa un recorrido completo del algoritmo de entrenamiento a lo largo de todo el conjunto de entrenamiento, por lo que debe definirse el número de épocas a utilizar para el entrenamiento, lo que puede ayudar a mejorar la precisión del detector.

El mini lote define un subconjunto del conjunto de entrenamiento que se debe utilizar para evaluar el gradiente de la función de pérdida y actualizar los pesos, por lo que el parámetro mini lote define el tamaño que se desea usar para cada iteración de entrenamiento y este solamente se puede representar con un número entero positivo. La

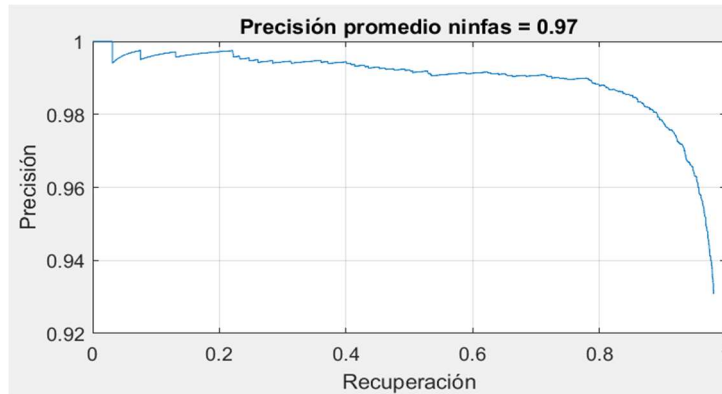


Fig. 9. Gráfica de precisión promedio al detectar ninfas de la mejor configuración.

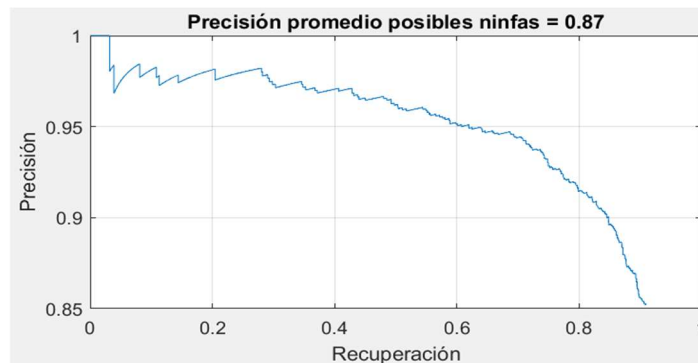


Fig. 10. Gráfica de precisión promedio al detectar posibles ninfas de la mejor configuración.

tabla 2 muestra las configuraciones que se exploraron para entrenar el sistema inteligente detector de ninfas de mosca blanca presentes en hojas de plantas.

Detector

El entrenamiento de la red entrega un objeto que contiene los parámetros de la red entrenada, aquí referido como “detector”. Para validar el correcto entrenamiento del detector se pone en operación al procesar una imagen de referencia, en la Fig. 7 se muestra un ejemplo de funcionamiento correcto de la red entrenada.

En la Fig. 8 se puede observar que en la imagen de referencia se insertan los cuadros delimitadores obtenidos con la información como las cajas delimitadoras de la región de interés (x, y, ancho, alto) y el porcentaje de precisión.

4. Validación y análisis de precisión

El enfoque para validar este sistema fue mediante la configuración de la red neuronal cambiando los parámetros de épocas, cajas de anclaje y los mini lotes de entrenamiento.

Tabla 2. Configuraciones para entrenar la red.

Redes	Cajas de anclaje	Épocas	Mini lote
Red 1	5	60	2
Red 2	25	60	2
Red 3	50	60	2
Red 4	5	20	2
Red 5	5	40	2
Red 6	5	60	8
Red 7	5	60	16
Red 8	25	80	2
Red 9	25	90	2
Red 10	25	80	8

Tabla 3. Precisión promedio al detectar candidatas a ser ninfas de múltiples redes.

Red	Precisión promedio de ninfas	Precisión promedio de ninfas incompletas	Precisión promedio de posibles ninfas
Red 1	96	90	79
Red 2	96	91	83
Red 3	96	91	83
Red 4	93	74	66
Red 5	95	84	79
Red 6	96	90	79
Red 7	94	85	73
Red 8	97	92	87
Red 9	97	93	89
Red 10	97	91	85

Se analizó la precisión obtenida en la detección de las candidatas a ser ninfas. Se obtuvieron la precisión, recuperación y precisión promedio de acuerdo con la Ecuación 2, donde VP representa los verdaderos positivos, FP representa los falsos positivos y FN los falsos negativos:

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

$$\text{Recuperación} = \frac{VP}{VP+FN} \quad (2)$$

$$\text{Precisión Promedio} = 2 \times \frac{\text{Precisión} \times \text{Recuperación}}{\text{Precisión} + \text{Recuperación}} \quad (3)$$

Se obtuvo la precisión promedio al utilizar cada uno de los detectores obtenidos de las configuraciones que se muestran en la tabla 2 y se recopiló la información de los resultados en la tabla 3. Al realizar pruebas con varios valores para los parámetros de cajas de anclaje, épocas y mini lote de entrenamiento se deduce la configuración del

sistema inteligente que arroja los resultados con mejor precisión promedio al detectar a las ninfas de mosca blanca presentes en hojas.

La red 9 cuenta con 150 capas y se configuró con 25 cajas de anclaje, 90 épocas y 2 mini lotes de entrenamiento, obteniendo precisiones que van desde 93% hasta un 97% al detectar candidatas a ser ninfas. En la Fig. 9 y 10 se muestran las gráficas de precisión promedio para la detección de las clases de objetos definidas.

A pesar de que los resultados que se reportan en esta sección fueron obtenidos utilizando un banco de imágenes que contiene exclusivamente hojas de tomate infestadas, la metodología propuesta puede utilizarse con imágenes de hojas de diferentes plantas infestadas de mosca blanca. En tal caso, deberá generarse una ecuación de compensación cromática diferente, así como realizar un nuevo entrenamiento de la red.

5. Conclusiones

El propósito del trabajo reportado es la detección de ninfas de mosca blanca sobre hojas de plantas, para su validación se utilizó un banco de imágenes de 224x224 píxeles de hojas de tomate infestadas con ninfas de mosca blanca.

De las diferentes pruebas de validación realizadas, se determinó que la configuración adecuada para entrenar la red consiste en utilizar 25 cajas de anclaje, 80 épocas de entrenamiento y un tamaño de mini lote de 2 para obtener los resultados con precisiones mayores. Se lograron precisiones que van del 93% al 97%, en la detección de ninfas nítidas.

En contraste, en la detección de ninfas que se ven borrosas y/o presentan formas irregulares se lograron precisiones que van desde 66% hasta 89%, en este caso, el entrenamiento y, en consecuencia, la detección, no se logra de manera eficaz debido a la falta de información característica de las ninfas en los objetos a detectar.

Por lo que, se puede concluir que si se cuenta con imágenes nítidas de las ninfas sobre hojas de plantas se pueden lograr detecciones con alta precisión al utilizar un sistema inteligente basado en la red neuronal YOLO V2.

Referencias

1. Organización de las naciones unidas para la agricultura y la alimentación. Agricultura y diálogos de cultura (2005) www.fao.org/home/es
2. El estado mundial de la agricultura y la alimentación (2021) www.fao.org/documents/card/es/c/cb4476es
3. DGSV-CNRF: Mosquita blanca Bemisia Tabachi (Gennadius). Estado de México (2020)
4. Elena-Cuéllar, M., Morales-Francisco J.: La mosca blanca Bemisia Tabaci (Gennadius) como plaga y vectora de virus en frijol común (Phaseolus vulgaris L.). Revista Colombiana de Entomología, vol. 32, no.1 (2006)
5. Bernal, L., Pesca, L., Rodríguez, D., Cantor, F., Cure, J. R.: Plan de muestreo directo para Trialeurodes vaporariorum (Westwood) (Hemiptera: Aleyrodidae) en cultivos comerciales de tomate. Agronomía Colombiana, vol. 26, no. 2, pp. 266–276 (2008)
6. Legg, J.: Bemisia tabaci: The whitefly vector of cassava mosaic geminiviruses in Africa: An ecological perspective. African Crop Science Journal, vol. 2, no. 4, pp. 437–448 (1994)

7. Gómez-Camperos, J. A., Jaramillo, H. Y., Guerrero-Gómez, G.: Técnicas de procesamiento digital de imágenes para detección de plagas y enfermedades en cultivos: Una revisión. *Ingeniería y Competitividad*, vol. 24, no. 1 (2022) doi: 10.25100/iyc.v24i1.10973
8. DASH Technologies: Towards Future Farming: AI is transforming the agriculture industry (2021) dashtechinc.com/towards-future-farming-ai-is-transforming-the-agriculture-industry/
9. Lino-Attyla, F. S., Silva-Brunna, C. R., Rocha-Danilo, P. C., Furriel-Geovanne, P., Calixto-Wesley, P.: Performance of haar and LBP features in cascade classifiers to whiteflies detection and counting. In: CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies, pp. 1–6 (2017) doi: 10.1109/chilecon.2017.8229737
10. Bernal-López, O. F., Medina-Melendrez, M. G.: Sistema de captura y procesamiento digital de imágenes para el conteo de mosca blanca en estado ninfal sobre hojas de tomate. Tesis de Maestría (2020)
11. Organización de las naciones unidas para la agricultura y la alimentación. Can artificial intelligence help improve agricultural productivity? (2017) www.fao.org/e-agriculture/news/can-artificial-intelligence-help-improve-agricultural-productivity
12. Gómez-Camperos, J., Jaramillo, H., Guerrero-Gómez, G.: Técnicas de procesamiento digital de imágenes para detección de plagas y enfermedades en cultivos: una revisión. *Ingeniería y Competitividad*, vol. 24, no. 1 (2021) doi: 10.25100/iyc.v24i1.10973
13. Liakos, K., Busato, P., Moshou, D., Pearson, S., Bochtis, D.: Machine learning in agriculture: A review. *Sensors*, vol. 18, no. 8, pp. 2674 (2018) doi: 10.3390/s18082674