

# Implementación de un Algoritmo para el Conteo de Aves Automatizado en el Sector Avícola

**Ryan A. LEÓN LEÓN**

Ingeniería, Universidad Privada del Norte  
Trujillo, La Libertad, Perú

**Hosmar Y. NEGREIROS ALONZO**

Ingeniería, Universidad Privada del Norte  
Trujillo, La Libertad, Perú

**Dilver AGUILAR ARTIAGA**

Ingeniería, Universidad Privada del Norte  
Trujillo, La Libertad, Perú

**Claudio C. PONCE LEÓN**

Ingeniería, Universidad Privada del Norte  
Trujillo, La Libertad, Perú

**Erick SALINAS POLO**

Ingeniería, Universidad Privada del Norte  
Trujillo, La Libertad, Perú

**Fabian A. CHÁVEZ TRELLES**

Ingeniería, Universidad Privada del Norte  
Trujillo, La Libertad, Perú

## **Implementation of an Algorithm for Automated Bird Counting in the Poultry Sector.**

**ABSTRACT:** *The poultry industry faces challenges in production control of bird counting. Artificial vision algorithms improve operational efficiency and animal welfare. These systems allow automated bird counting, reducing time. Python is key in its development, also facilitating the creation of interfaces and the optimization of algorithms. This experimental research collected data using a 720px Webcam and a Core i3 laptop. YOLOv8 was used to create the neural structure by capturing images of the birds, to perform image processing, the neural network was trained, evaluated, the system was implemented and concluded with machine learning. The results demonstrated an accuracy of 85.36% when detecting and counting birds. In conclusion, this technology is reliable and effective, with accurate bird counting, helping in the poultry industry process.*

**Keywords:** *Poultry automation, Counting algorithm, birds, AI Python, yolov8.*

**RESUMEN:** La industria avícola enfrenta desafíos en el control de producción del conteo preciso de aves. Los algoritmos de visión artificial mejoran la eficiencia operativa y bienestar animal. Estos sistemas permiten un conteo automatizado de aves, reduciendo tiempos. Python es clave en su desarrollo, facilitando también la creación de interfaces y la optimización de algoritmos. Esta investigación de tipo experimental recopiló datos mediante una Cámara Web 720px y una laptop Core i3. Se utilizó YOLOv8 para realizar la estructura neuronal capturando imágenes de las aves, para realizar el procesamiento de las imágenes, se entrenó la red neuronal, se evalúa, se implementó el sistema y concluyó con el aprendizaje automático. Los resultados demostraron una precisión del 85.36% al detectar y contabilizar las aves. En conclusión, esta tecnología es confiable y eficaz, siendo exacto el conteo de aves, ayudando en el proceso de la industria avícola.

**Palabras Claves:** Automatización avícola, Algoritmo de conteo, aves, IA Python, yolov8.

## **1. INTRODUCCIÓN**

La industria avícola, a gran escala, enfrenta numerosos desafíos en la gestión y control de producción. Una de las problemáticas es la contabilidad precisa de las gallinas en granjas de grandes dimensiones. En este rubro de crianza de gallinas y producción de huevos a nivel mundial, China lidera con 39.1%, seguido por EE. UU. con 7.8% hasta el 2017, lo que nos da a entender que desde el año 2000 este rubro crece a ritmos gigantescos, dado que para el 2017 ha habido un incremento del 48.1%. Mientras tanto, en América, los países con mayor índice de crianza son México y Colombia. Por otro lado, el Sistema Integrado de Estadística Agraria del Perú nos da a entender que las regiones donde existe mayor índice de crianza al 2018 se concentran en Ica, Lima, La Libertad, Arequipa y San Martín [1]. En lo que respecta a la producción nacional de pollo, en diciembre de este año, las regiones productoras más destacadas están en Lima (57,0%), La Libertad (17,3%) y Arequipa (8,8%). Por un incremento en la colocación de pollos "bebé" de la línea de carne en estas regiones en comparación con el mismo periodo del año previo. La avicultura a nivel nacional se ubica principalmente en la costa y próxima a los principales centros de consumo del país, y posee una significativa contribución al Valor Bruto de la Producción Agropecuaria (26,3%). Se destaca como una actividad en expansión y brinda datos de la colocación reproductora de pollos Bb de engorde (66 209 621), pavosBb de engorde (367 207) y patos Bb de engorde (108,842) [2]. La carne de pollo es el alimento más asequible y el principal proveedor de proteínas para las familias en Perú, además de ser el componente principal de nuestra gastronomía. Igualmente, en Perú se consumieron 55 kg de carne de ave por persona en 2021 y, alrededor de 1 kg de carne de ave se consumía por semana. Además, es relevante tener en cuenta que Perú es el país con el mayor consumo individual de

carne de pollo en América Latina y se sitúa en el Top 4 global en términos de consumo individual de carne de pollo. Igualmente, el sector avícola crea 460 mil empleos, tanto directos como indirectos, garantizando de esta manera el sustento de más de 1.8 millones de individuos. En cambio, en términos económicos, el sector avícola aportó a la economía peruana el 65% del valor de la producción pecuaria, el 24% del valor de la producción agropecuaria y alrededor del 2% del PBI nacional, en el 2021 [3]. En tal sentido, se busca aplicar el desarrollo de algoritmos para el conteo de aves en el sector avícola, lo que representa una innovación clave en la automatización de procesos de gestión en la industria. Estos algoritmos no solo permiten optimizar la eficiencia operativa al reducir los tiempos y costos asociados al conteo manual, sino que también mejoran la precisión, lo que es lo más importante en instalaciones de gran escala. La automatización del conteo contribuye además al bienestar animal, ya que minimiza el estrés generado por la intervención humana directa, al permitir que el monitoreo se realice de forma no intrusiva. Estos sistemas avanzados pueden detectar patrones, comportamientos anómalos y hasta identificar posibles problemas de salud en las aves, lo que facilita una respuesta temprana y proactiva. Asimismo, la capacidad de generar datos en tiempo real permite una mejor toma de decisiones estratégicas en cuanto a la planificación de la producción, la gestión de inventarios, y la asignación de recursos, haciendo que la industria avícola sea más competitiva y sostenible en un entorno donde la demanda de productos avícolas sigue en crecimiento a nivel global [4]. El uso de técnicas de visión artificial demuestra ser una solución eficaz para enfrentar los desafíos de presión y rapidez en el conteo, en entornos avícolas donde las condiciones suelen ser cambiantes y complejas. La visión artificial, basada en algoritmos avanzados de procesamiento de imágenes y aprendizaje profundo, facilita la identificación y clasificación de las aves de manera no invasiva, minimizando la perturbación en su comportamiento natural. Además, estas tecnologías se pueden integrar con sensores adicionales, como cámaras térmicas o de profundidad, para aumentar aún más la precisión y la robustez del sistema en condiciones adversas. Esta capacidad de adaptarse a escenarios difíciles, como granjas al aire libre o instalaciones cubiertas con miles de aves en movimiento, ofrece a los productores avícolas una herramienta poderosa para mejorar la eficiencia, reducir los costos operativos, y garantizar un control de calidad más riguroso en todas las etapas del proceso productivo [5]. Se destaca también la importancia de desarrollar algoritmos robustos y adaptables que puedan enfrentar estos desafíos específicos mediante el uso de imágenes, asegurando precisión y rapidez en los conteos. Estos algoritmos deben ser capaces de operar en condiciones complejas y cambiantes, como variaciones en la iluminación, movimiento rápido de los animales y diferentes entornos. Además, su implementación no solo mejora la eficiencia operativa, sino que también permite una mejor toma de decisiones basada en datos confiables, reduciendo errores humanos y mejora los procesos de gestión en las empresas avícolas [6]. También, sugieren que el desarrollo de algoritmos específicos para la avicultura no sólo mejora la eficiencia operativa, sino que también contribuye al bienestar animal ya que permite a las apersonas monitorear con mayor exactitud el estado y la cantidad de aves presentadas en los galpones [7]. Al automatizar el proceso de conteo y monitoreo, se eliminan las imprecisiones inherentes a la intervención manual, especialmente en entornos donde el volumen de aves es alto y las condiciones pueden ser desafiantes. Estos sistemas permiten un seguimiento continuo y en tiempo real, lo que facilita la gestión eficiente de los recursos, mejora la trazabilidad, y optimiza la planificación operativa. [8].

La versatilidad de Python permite integrar fácilmente bibliotecas como OpenCV para el procesamiento de imágenes y ZBar o Pzbar para la lectura de códigos QR. Además, estas soluciones son escalables y de bajo costo, lo que las convierte en una opción ideal para pequeñas y medianas empresas que buscan optimizar sus procesos con tecnología accesible [9]. También cabe destacar que a lo largo del tiempo la visión artificial con realidad aumentada viene dando soluciones óptimas en las industrias con una creciente exponencial que incluso aborda altos costos de implementación [10].

Asimismo, el estudio de Wang y otros, 2023; señalan que la “Revisión de Métodos y Aplicaciones de Conteo Visual para Animales de Granja” En los últimos años, con el desarrollo de equipos, tecnología y algoritmos por computadora, la investigación sobre el conteo de animales basado en visión por computadora también ha logrado grandes avances. Los métodos de conteo basados en visión por computadora pueden realizar un conteo automatizado [11]. Pero no solo se aplica a este tipo de conteo sino también a otros objetos, como se menciona en la investigación “Sistema de Visión Artificial para Conteo de Objetos en Movimiento” En este artículo se hace una introducción a diferentes conceptos y metodologías utilizadas en la construcción y reconocimiento de patrones; así mismo, el objetivo de este tipo de sistemas es el de obtener información de interés, que se pueda interpretar, a partir de un complejo procesamiento de imágenes obtenidas por medio de dispositivos ópticos [12]. Por otro lado, el estudio de “Detección de Ciervos en Imágenes para Contar Poblaciones Mediante Métodos de Visión por Computadora” tiene como objetivo diseño y desarrollo de sistemas de inteligencia artificial; aumento de imágenes para tareas de visión por computadora; algoritmos de ajuste de hiper parámetros para entrenar modelos de redes neuronales [13]. Mientras que la investigación “Conteo de personas con un sensor RGBD comercial”, tiene como objetivo detectar y contar personas que cruzan una puerta en cualquier espacio sin restringir las condiciones de iluminación, además, se trataría de obtener lo mejor de ambos enfoques, basando el procesamiento en todos los canales de información alineados por los sensores RGBD de bajo coste. El método propuesto es robusto en escenarios de baja o media densidad de personas ante grandes o repentinos cambio de iluminación o ausencia de este. En el enfoque basado en gaussianas proporciona resultados ligeramente mejores [14]. En otra investigación se demuestra que la contabilización de animales con YOLO es eficaz para conocer la cantidad de producción en tiempo real [15].

El Objetivo general, es implementar un algoritmo para el conteo de aves automatizado en el sector avícola. [16]. Por tanto, como objetivos específicos, se desarrollará un conjunto de datos con imágenes de aves de corral que permita entrenar y evaluar el conteo con Roboflow. Además, desarrollar una red neuronal con YOLO8v y Python que identifique y automatice el conteo de aves en tiempo real mediante una cámara. A esto también, se evaluará la precisión y efectividad del algoritmo propuesto durante el entrenamiento e implementación [17].

## 2. MÉTODOS Y MATERIALES

En esta investigación de tipo experimental se desarrolló un algoritmo que facilitó el conteo de aves con precisión y eficiencia. Como población se tuvo las aves de diferentes especies en una granja avícola, seleccionando algunas especies como muestra. Para el desarrollo del algoritmo, se utilizó

imágenes capturado con una Cámara Web 720p y una laptop para registrar el número de aves (ver **Figura 1**), luego, se pre-procesó las imágenes usando técnicas para mejorar la calidad de las grabaciones, aplicamos redes neuronales convencionales (CNN) como técnica de aprendizaje automático del algoritmo YOLOv8, optimizador AdamW (lr=0.000714, momentum=0.9) con los grupos de parámetros 26 weight (decay=0.0), 27 weight (decay=0.0005), 27 bias (decay=0.0). AdamW **Ec. (1)** que ayudó a mantener la estabilidad y el rendimiento del modelo evitando el decaimiento de los pesos, para lograr una gran precisión.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (1)$$

La regularización L2 es una técnica que fue utilizada para evitar el sobreajuste añadiendo un término de penalización a la función de pérdida, desalentando los valores de peso grandes que ayudó a crear modelos más sencillos que se generalizaron mejor a los nuevos datos **Ec. (2)**.

$$v_t = \beta_2 v_{t-1} + (1 + \beta_2) g_t^2 \quad (2)$$

Aplicar el decaimiento **Ec. (3)** del peso de forma independiente resolvió uno de los principales defectos de Adam que es su tendencia a afectar a las actualizaciones del gradiente durante la regularización, la ventaja clave fue mejorar su rendimiento, sobre todo en escenarios de modelado complejos.

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} - \eta \lambda \theta_t \quad (3)$$

Con la estimación de corrección de sesgos **Ec. (4)** se utilizó el optimizador Adam, que es más eficaz en la optimización y contribuyó a mejorar la formación del modelo y su robustez.

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (4)$$

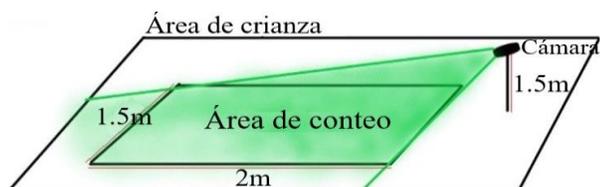


Figura 1. Diseño y posicionamiento para el conteo.

**Arquitectura del modelo neuronal con YOLOv8:** La arquitectura de YOLOv8, el cual muestra la entrada de diferentes imágenes de aves en tiempo real mediante una cámara, en la capa de convolución se realizó varios tipos de transformaciones a las imágenes como cambio de tamaño o posición, mientras que en la extracción de características clasifica bordes, texturas, patrones complejos. Se optó por la función de activación Leaky ReLU **Ec. (5)** en sus capas convolucionales debido a varias razones, Leaky ReLU evita el problema de las "neuronas muertas" que se presenta en redes con ReLU estándar, donde las neuronas no pueden actualizar sus pesos cuando la activación es cero.

$$f(x) = \{x, \alpha x, x \geq 0; 0, x < 0\} \quad (5)$$

Leaky ReLU permite un pequeño gradiente negativo (con un parámetro  $\alpha = 0.01$ ), lo cual mejoró la capacidad de aprendizaje de la red, especialmente en las capas profundas. A diferencia de otras funciones como (e.g., ReLU y GELU), Leaky ReLU es menos costosa computacionalmente, lo que la hace más adecuada para tareas en tiempo real. Por otro lado, en las predicciones de

cajas delimitadoras encierra el objeto **Ec. (6)**, iniciando con 32 filtros el cual entregó a las capas de salida un total de 255 filtros en las imágenes de las aves etiquetadas con el puntaje de confianza y sumando el contador, así mismo, utilizó el proceso Non-Max Suppression para eliminar cajas duplicadas y que el entrenamiento tenga velocidad, precisión, eficiencia de recursos, capacidad de detección en variación de distancia con ángulo de visión diferentes.

$$x = \sigma(t_x) + C_x; y = \sigma(t_y) + C_y; w = p_w e^{t_w}; h = p_h e^{t_h} \quad (6)$$

Por lo tanto, se decidió trabajar con la versión YOLOv8, porque a comparación de versiones anteriores (YOLOv4, YOLOv5, YOLOv7) presenta una mayor precisión, velocidad y versatilidad en la detección de objetos, siendo importante para el alcanzar el objetivo del proyecto.

Principalmente la elección de YOLO para aplicar en esta investigación fue por los recursos computacionales que no es tan exigente a diferencia de CNN, dado que nos limitaban los recursos para el desarrollo e implementación, además la velocidad de funcionamiento es alta y más fácil de implementar a comparación de otros algoritmos (ver **Tabla 1**).

Tabla 1. Compartivo con otros algoritmos

Criterio	YOLOv8	Faster R-CNN	SSD
Velocidad (FPS)	Muy alta (80-120 FPS en GPU moderna)	Baja (5-15 FPS en GPU moderna)	Alta (30-60 FPS en GPU moderna)
Precisión mAP	Alta (70-75%)	Muy alta (75-80%)	Media-Alta (60-70%)
Ef. tiempo real	Óptima	Poco eficiente	Moderada
Recursos	Moderados	Altos	Moderados
Implementar	Sencilla	Complejo	Sencilla
Robustez	Buena	Excelente	Media
Tamaño del modelo	Compacto (~50 MB)	Muy grande (>200 MB)	Compacto (~100 MB)

En el siguiente diagrama (ver **Figura2**), se detalla el desarrollo completo de la investigación basado en la captura de imágenes en tiempo real.

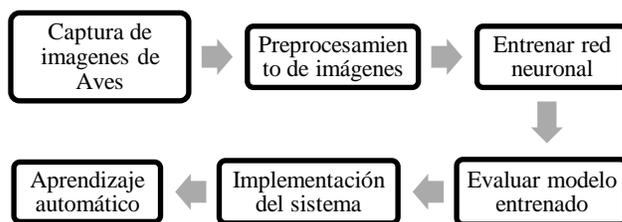


Figura 2. Diagrama para el desarrollo de la investigación.

**Captura de imágenes de aves:** Para el registro visual de las aves, se capturaron imágenes utilizando cámaras de celulares Huawei de 8 MP y Xiaomi Redmi de 13 MP, con una distancia focal de 1x, asegurando el encuadre adecuado de las aves en movimiento. Se consideraron múltiples ángulos de captura, abarcando desde 0° (vista frontal) hasta 90° (vista cenital), con el fin de obtener una variedad de perspectivas y mejorar la robustez del modelo (ver Figura 3). Asimismo, se tomaron imágenes en diferentes entornos, incluyendo galpones cerrados y espacios abiertos, con el objetivo de evaluar el desempeño del sistema en diversas condiciones de iluminación y fondo. Para ampliar la diversidad

del conjunto de datos, se realizó una búsqueda y recopilación de imágenes de alta resolución en la web, asegurando que estas fueran de dominio público o contaran con los derechos de uso adecuados. Adicionalmente, se emplearon técnicas de preprocesamiento de imágenes, como el ajuste de brillo y contraste, reducción de ruido y escalado de resolución, con el fin de estandarizar las características visuales y optimizar el rendimiento del modelo. También se exploró el uso de cámaras con sensores infrarrojos y visión nocturna, con el propósito de evaluar su viabilidad en la detección de aves bajo condiciones de baja iluminación o en horarios nocturnos.

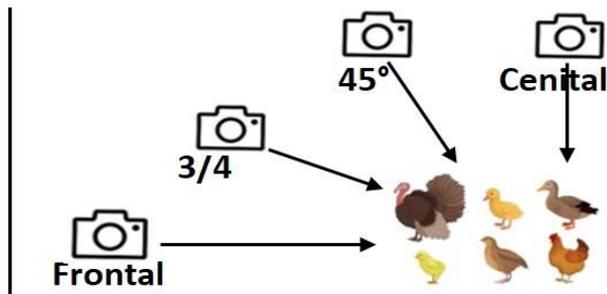


Figura 3. Detalle de ángulos para captura de fotografías.

**Preprocesamiento de imágenes:** El proceso inició con la recopilación de 204 imágenes de aves en formato digital, siendo 104 fotografías capturadas y 100 obtenidos de la web el cual es la base suficiente para detectar las primeras aves, dado que el modelo seguirá reentrenando al estar implementado en la realidad. Estas fotografías se mejoraron con Roboflow, únicamente aquellas que presentaban malas condiciones como iluminación baja, movidas, difuminadas, borrosas, entre otros; dichas imágenes no solo se aplicó mejoras en la forma visual sino también en la rotación. A continuación, se aplicó un procesamiento de imagen en Roboflow que incluyó la mejora de contrastes de fotografías lo que permitió resaltar características importantes del objeto para su posterior análisis. Adicionalmente, se etiqueta los objetos de manera rectangular o poligonal libre con la clase bird permitiendo identificar las aves según criterios (ver Figura 4). Una vez que la imagen ha sido segmentada es posible identificar los objetos presentados agrupando los píxeles que comparten la misma etiqueta y que a la vez sean adyacentes con las regiones u objetos etiquetados se pudo avanzar hacia las sub-clasificación o reconocimiento automático permitiendo extraer características discriminantes y obtener información relevante sobre los objetos [18].

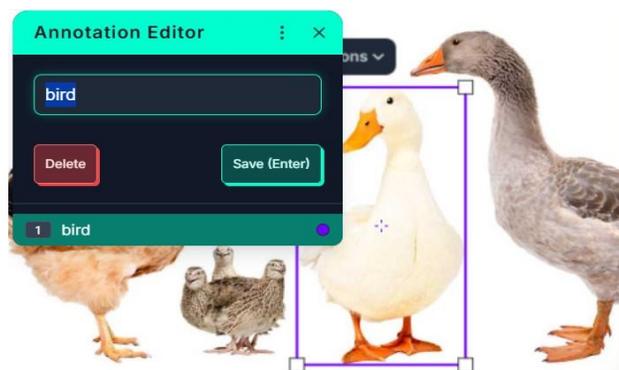


Figura 4. Etiquetado de objetos con la clase bird.

Al culminar las anotaciones se procedió agregar las 204 imágenes al dataset el cual se divide automáticamente en categorías de train

set(74%) con 151, valid set(18%) con 37 y test set(8%) con 16 imágenes. Finalmente se agregó aumento de voltear horizontal para objetos que aparecen reflejados, con esto, se evita tomar una sola etiqueta en dos aves posicionadas de cola que sea detectada como una sola y rotación entre  $-20^\circ$  a  $20^\circ$  que simula los ángulos de posición de las aves en dicho rango, por lo tanto, no se puede exceder de esos valores para no distorsionar la relevancia de los objetos sobre todo en un entorno que no es rotacionalmente invariante. Lo expuesto se aplicó para cada imagen donde, el modelo aprende a generar versiones aumentadas en su grupo de entrenamiento el cual se obtuvo un total de 506 imágenes con el siguiente conjunto de datos train set(90%) con 453, valid set(7%) con 37 y test set(3%) con 16 imágenes. **Entrenamiento de red neuronal:** Para poder entrenar la red neuronal se siguió el siguiente flujo (ver Figura 5).

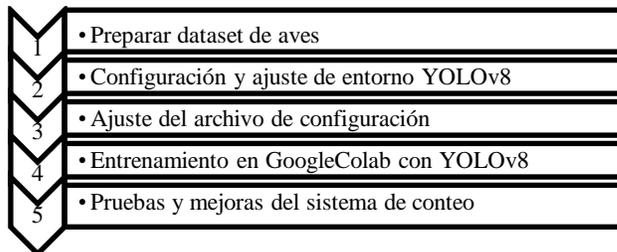


Figura 5: Fases de entrenar la red neuronal con YOLOv8

El dataset de las aves necesitó un conjunto de imágenes en entornos donde las aves se encuentren en galpones, corrales o campo de pastoreo. Se priorizó la captura de imágenes en diversas condiciones de iluminación y ángulos para mejorar la robustez del modelo. Además, se incluyeron diferentes especies y estados de crecimiento para garantizar una mayor generalización en la detección y clasificación, con diferentes condiciones de luz, ángulos y distancias. También fue necesario capturar las especies en diferentes posiciones, dado que para YOLOv8 es necesario contar con las coordenadas del centro de caja, ancho y alto de la caja normalizado con una clase birds asignadas [19]. En un archivo YAML se configuró los ajustes necesarios más las rutas de los datos de entrenamiento, validación y la clase bird, para proceder con el entrenamiento YOLOv8 con un código simplificado [20]. Validamos el modelo realizando las pruebas y aplicando las mejoras con la detección y conteo en tiempo real con OpenCV que cuenta las aves en cada fotograma [21]. Para terminar, se ajustó la tasa de aprendizaje aumentando el tamaño de lote y ajustamos el tamaño de caja que cubran bien todas las aves y ser más certero. Para prevenir el sobreajuste y mejorar la capacidad de generalizar del modelo, se utilizó la técnica de Normalización de Batch, que se integra en las capas convolucionales del modelo YOLOv8. Esta técnica permitió estabilizar las activaciones neuronales en cada minibatch, asegurando un aprendizaje más eficiente. Además, se complementó con estrategias como el aumento de datos (rotaciones, escalado y ajustes de brillo/contraste) y el ajuste de parámetros, incluyendo un tamaño de lote de 32 con un batch=16, una tasa de aprendizaje predeterminada de roboflow del 0.01 y una regularización L2 (weight decay) de 0.005, también mencionar que durante el entrenamiento con GoogleColab, se adicionó los parámetros task=detect por ser de tipo detección, mode=train para que entrene el nuevo modelo, partiendo como base del modelo yolov8n.pt y combinándolo con el dataset data.yaml el cual contiene las etiquetas que fue trabajado en Roboflow, además se establece las epochs=100 que representa la cantidad de veces a entrenar, batch=16 el tamaño de lote para a procesar juntos antes de actualizar parámetros, plots=True para

que nos genere gráficos como de interpretación como curvas y matrices, por ultimo con el parámetro `imgsz=640` redimensionamos las imágenes para evitar mayor consumo de memoria durante el entrenamiento el cual tardó 5.008 horas en completar. **Evaluar modelo entrenado:** A través de una red neuronal entrenada, el software analizó cada cuadro de video en tiempo real y localizó las aves dentro del campo de visión de la cámara el cual define que es una “detección correcta” para lo cual se analiza el concepto de intersección sobre la unión con métrica de evaluación más popular utilizada en los puntos de referencia de detección de objetos en el ámbito de la detección, el IOU Ec. (7), midió el área de superposición entre el cuadro delimitador predicho y el cuadro delimitador real que debía ser detectado (ground truth) dividido por el área de unión entre ellos con las predicciones correctas, falsos positivos que se debe al etiquetado rectangular que sobresale de la zona del objeto el cual lo compara con el ambiente el cual obtiene como resultado un ave errada, falsos negativos es porque los parámetros aplicados no fueron suficiente para arreglar la iluminación así como con la condición de los ángulos de las fotografías, además nos muestra los negativos correctos (ver **Figura 6**). Por tanto, el rendimiento de precisión en la **Ec. (8)** fue de 68%, la sensibilidad o recall en la **Ec. (9)** fue de 68% y el F1-score de 67.9% **Ec. (10)**.

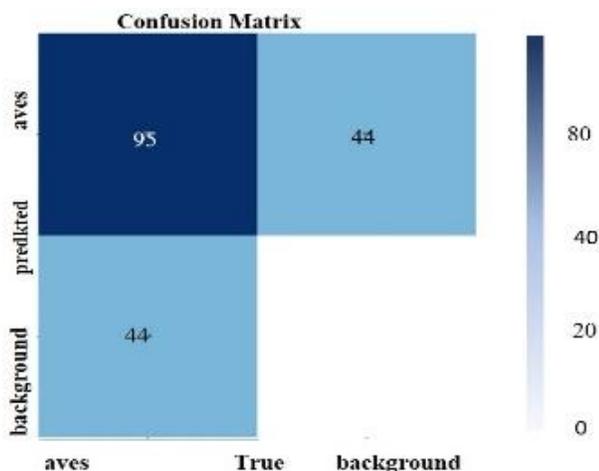


Figura 6. Matriz de confusión

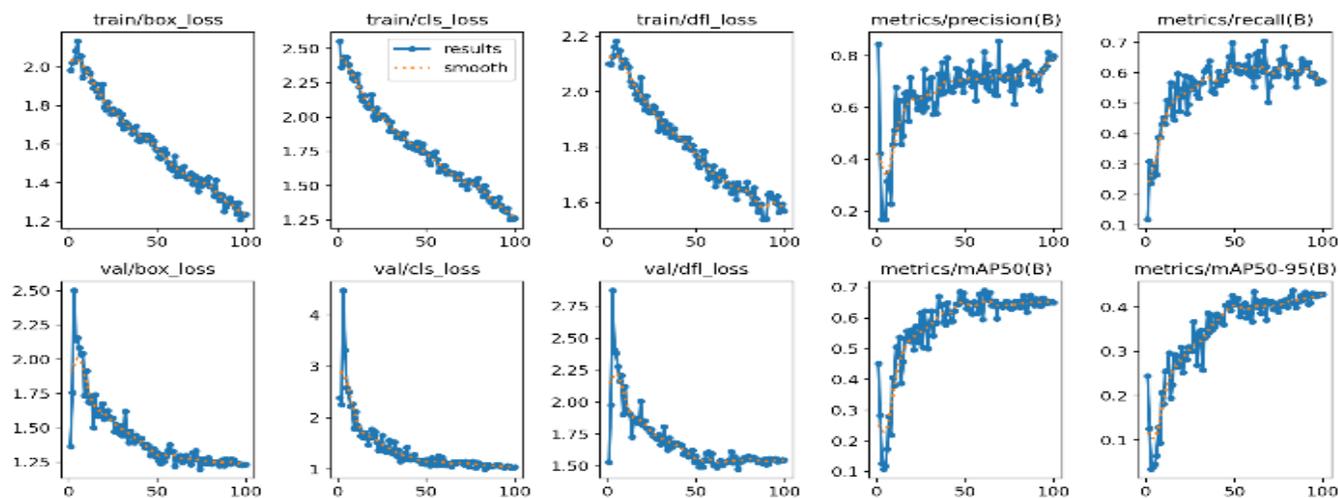


Figura 7. Resultados del entrenamiento

En el gráfico de desempeño (ver **Figura 8**) se muestran las distintas métricas de entrenamiento y validación del modelo a lo

$$IOU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} \quad (7)$$

**Dónde:**  $B_p$  es el cuadro predicho y  $B_{gt}$  es el cuadro de ground truth. Al comparar el IOU con un umbral dado, se considera que una detección es correcta si IOU mayor o igual que dicho umbral e incorrecta en caso contrario.

$$Precisión = \frac{True\ Positivos\ (VP)}{True\ Positivos\ (VP) + False\ Positiv\ (FP)} \quad (8)$$

La sensibilidad o también conocida como recall, mide la capacidad del algoritmo para detectar todas las aves presentes. Es decir, de todas las aves que están realmente en la escena, cuántas fueron detectadas.

$$Recall = \frac{True\ Positivos\ (VP)}{True\ Positivos\ (VP) + Flase\ Negativos\ (FN)} \quad (9)$$

El F1-score refleja un balance entre la precisión y recall para la clase aves o birds.

$$F1 - score = 2 * \frac{Precisión * Recall}{Precisión + Recall} \quad (10)$$

En el gráfico de resultados (ver **Figura 7**) se muestra una reducción progresiva en las pérdidas de caja, clasificación y distribución `box_loss` que es la pérdida asociada a la precisión en la ubicación de los cuadros delimitadores bounding boxes, `cls_loss` es la clasificación de qué tan bien el modelo distinguió las categorías, `dfl_loss` relacionada con la detección de bordes, lo que indicó que si aprendió correctamente. A la vez, las métricas de precisión, recall y mAP mostraron un crecimiento progresivo, que refleja una mejora en la detección de objetos. No se observaron signos de sobreajuste, ya que las pérdidas de validación siguieron una tendencia similar en el entrenamiento. En general, los resultados obtenidos del modelo apuntaron a una mejora en el rendimiento de manera estable a lo largo de las 100 épocas establecidas.

largo de las 100 épocas establecidas. Las curvas de evaluación de un modelo de detección de objetos para la clase "bird",

incluyendo F1-Confianza, Precisión-Confianza, Precisión-Recall y Recall-Confianza. La mejor relación entre precisión y recall se alcanzó con un umbral de confianza de 0.472 (F1 = 0.67). La precisión fue máxima (100%) cuando la confianza fue de 0.941, pero esto redujo la cantidad de detecciones. La curva de

Precisión-Recall indica un mAP@0.5 de 0.674, reflejando un rendimiento moderado. Finalmente, el recall disminuyó a medida que aumentó la confianza, con un máximo de 0.85 cuando la confianza es 0.

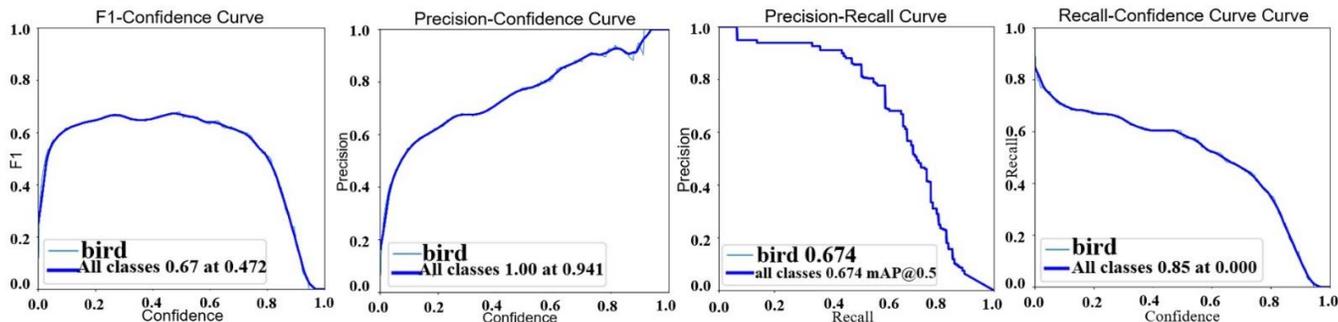


Figura 8. Métricas de desempeño

Durante el entrenamiento del modelo se utilizó una tasa de aprendizaje de  $lr=0.000714$ , con este valor se logró un balance adecuado entre la velocidad de convergencia y la estabilidad del aprendizaje, garantizando actualizaciones consistentes. Se buscó evitar oscilaciones en el gradiente que pudieran derivar en una pérdida de precisión CIoU Ec. (11), a la vez que se aseguraba una convergencia eficiente. Además, este valor se validó mediante experimentos previos que evaluaron su impacto en métricas de rendimiento como la precisión, la sensibilidad y el tiempo de convergencia, demostrando un equilibrio entre estos factores. Lo que evidenció que la elección del lr contribuye a un rendimiento confiable y consistente. Una tasa de aprendizaje mayor habría incrementado las fluctuaciones del gradiente, resultando en una posible divergencia del modelo, mientras que una tasa menor habría alargado innecesariamente el tiempo de entrenamiento. La selección de este valor equilibró adecuadamente la precisión y la sensibilidad del modelo, optimizando su capacidad de generalización (ver Figura ).

$$LCIoU = 1 - IoU + c^2 \rho^2(b, bg) + av \quad (11)$$

Donde: Pérdida LCIoU combina la superposición (IoU), la alineación de centros  $c^2 \rho^2$  y la similitud de forma  $av$  para mejorar la precisión en la detección de objetos, ayudando a reducir errores en la ubicación y el tamaño de las cajas predichas.

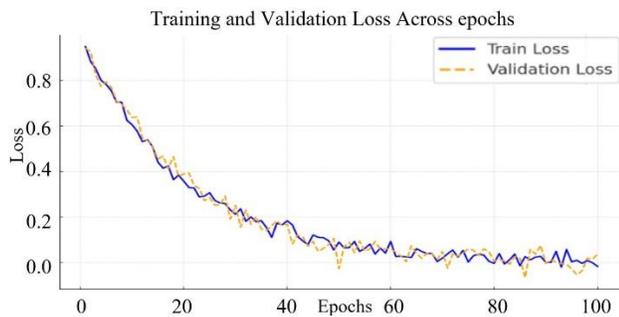


Figura 9. Función de pérdida a lo largo de las épocas

Así mismo, el resultado del etiquetado muestra cómo el modelo ha sido entrenado para detectar aves en diferentes entornos y con diversos niveles de confianza donde no se evidencia falsos positivos visibles (ver Figura 10). Las detecciones no muestran puntajes porcentuales por ser solo cajas delimitadoras bounding boxes que identifica y rodea la presencia de un ave, por otro lado, la precisión del modelo fue precisa porque las aves están bien

encuadradas dentro de los rectángulos azules. Esta visualización es útil para evaluar el rendimiento del modelo e identificar áreas donde el modelo podría necesitar mejorar como en la detección de aves en entornos complejos con fondo similar al ave o también ajustar la confianza mínima dado que algunas detecciones puedan solaparse con la misma especie u otros objetos.

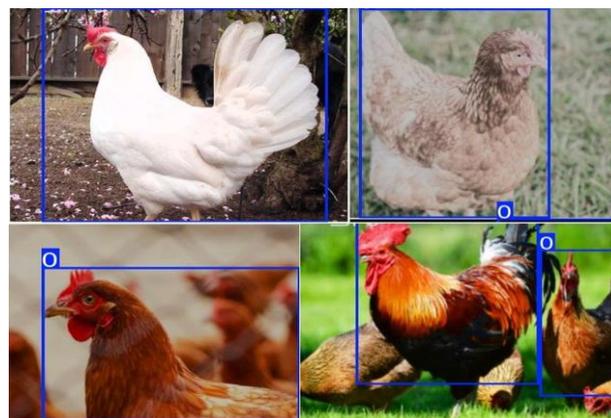


Figura 10. Etiquetado de aves.

**Implementación.** Este algoritmo se implementó en un entorno de desarrollo que permitió la integración de una cámara de video en tiempo real y el procesamiento de imágenes a través de YOLOv8 [22]. Para iniciar, se ejecutó el archivo principal del programa, el cual activa los módulos encargados de la captura de imágenes, la detección de objetos y el conteo de aves mostradas. Se aseguró que el sistema esté correctamente configurado con las dependencias de Yolo8v, OpenCV que sirve para capturar las imágenes reales, Ultralytics para detectar los objetos, librería OS para interactuar con el sistema operativo y librería Time para establecer tiempo para capturar imágenes. Además, detallamos los equipos que se emplearon en la implementación del conteo de aves en tiempo real (ver Tabla 2).

Tabla 2. Especificaciones del hardware

Equipo	Propiedades
Laptop	Procesador Intel Core i3, RAM 8 GB, x64bits, SO Windows 10, pantalla de 15.6"
Cámara Web	720mp

#### Contabilizar las aves

Para poder contabilizar las aves nos basamos en la Ec. (12),

donde recorre todos los objetos detectados en la imagen (desde  $i=1$  hasta  $i=m$ ) y suma 1 cada vez que el modelo clasifica el objeto como un ave. Si el objeto no es clasificado como un ave, no se suma al conteo. Esto dio el número total de aves presentes en la imagen [23].

$$N(aves) = \sum_{i=1}^m \delta(C_i) \quad (12)$$

$N(aves)$  es el total de aves contadas dentro de la imagen,  $C_i$  es la clasificación de cada ave detectado por el modelo,  $\delta(C_i)$  toma el valor 1 si es ave o cero sino lo es y  $m$  es el número total de aves detectados en la imagen (ver **Figura 21**).

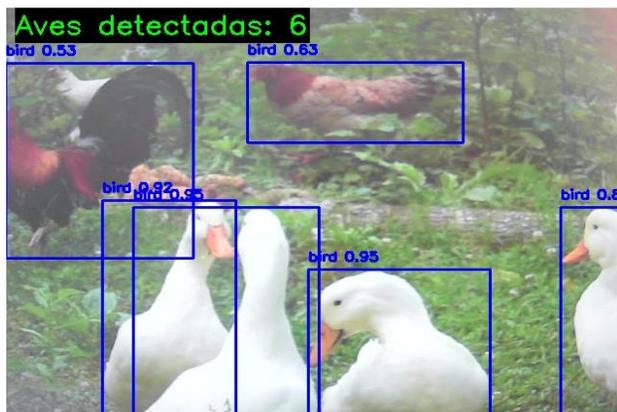


Figura 21. Detección en tiempo real

#### Validación de condiciones reales

Para la ubicación de las pruebas seleccionamos una granja representativa que permita trabajar en diferentes entornos como galpón cubierto, área de alimentación, etc. Los equipos utilizados fueron una cámara con especificaciones adecuadas con un mínimo de 720p y el software de detección basado en YOLOv8, instalado en una computadora de 8GB de RAM y un procesador Intel Core i3. La duración de la prueba se realizó al menos entre 5 a 7 días, cubriendo un ciclo completo de actividades diarias y condiciones de luz amanecer, anochecer, horas de máximo y mínimo de luz. Así mismo, designamos a un grupo humano para monitorear el sistema durante las pruebas, recolectar datos y realizar ajustes según sea necesario.

**Escenarios:** Conteo en el día bajo luz natural directa, con lluvia y nublado. También se validó en galpones cerrados con iluminación artificial constante, luz infrarroja de las criadoras. En condiciones de poca luz y sombras tales como al amanecer, al atardecer, y áreas de sombra intensa. Por otro lado, en los escenarios de comportamiento se validó la actividad intensa, por ejemplo, cuando las aves son alimentadas o corretean en el ambiente y analizar los momentos donde las aves tienden a agruparse.

**Indicadores de éxito y criterios de evaluación:** La precisión debe ser superior al 90% en todos los escenarios bien entrenados. La variación de precisión entre condiciones lumínicas distintas no debe ser mayor al 5%. Sensibilidad ante el comportamiento de las aves para mantener el conteo preciso debe ser de  $\pm 3\%$  en situaciones de alta densidad y movimiento [24]. Durante el desarrollo del modelo de detección y conteo de aves basado en YOLOv8, se presentaron varios desafíos técnicos. Uno de los principales retos fue la variabilidad en las condiciones de iluminación dentro de los galpones, lo que afectaba la detección

precisa de las aves. Para abordar este problema, se implementaron técnicas de aumento de datos como ajustes en el brillo y contraste, así como la normalización de las imágenes capturadas. Otro desafío importante fue la superposición de aves en entornos con alta densidad poblacional. Se resolvió mediante la optimización de los parámetros de Non-Max Suppression (NMS) para evitar detecciones duplicadas y mejorar la precisión del conteo. Asimismo, la selección de una tasa de aprendizaje adecuada fue crítica para evitar el sobreajuste del modelo, lo que se logró con el ajuste progresivo del learning rate y la utilización de la regularización L2 en el entrenamiento.

### 3. RESULTADOS

La implementación del algoritmo de conteo automatizado basado en YOLOv8 permitió reducir significativamente los tiempos de registro y monitoreo de aves en comparación con los métodos manuales utilizados tradicionalmente en la industria avícola. Al minimizar la intervención humana en el proceso de conteo, se optimiza la gestión operativa y se mejora el bienestar animal al reducir el estrés generado por la manipulación directa. Además, la capacidad del modelo para operar en entornos con variaciones lumínicas y distintas configuraciones espaciales permitió su aplicación en diferentes tipos de granjas avícolas, desde galpones cerrados hasta áreas abiertas. Esto representa un avance significativo en la automatización del sector, facilitando la toma de decisiones en la planificación de la producción y la gestión de inventarios con datos más precisos y en tiempo real.

#### Precisión Real de la Implementación

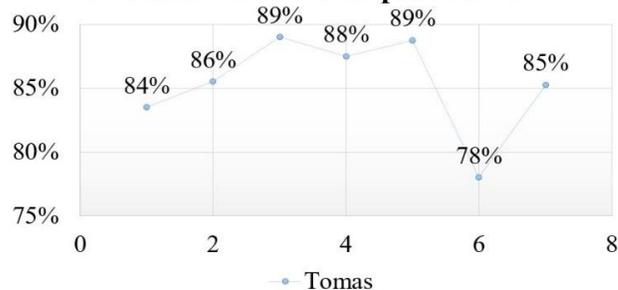


Figura 32. Tomas de la implementación.

YOLOv8 sobresale entre las redes neuronales diseñadas para detección de objetos gracias a su excepcional equilibrio entre rapidez, precisión y eficiencia. Este modelo es capaz de analizar imágenes en tiempo real, procesando hasta 35 cuadros por segundo con un tiempo promedio de inferencia de solo 0.028 segundos. Esto lo convierte en una solución ideal para aplicaciones críticas como vigilancia o monitoreo de espacios públicos, superando notablemente a arquitecturas como SSD y Faster R-CNN, cuyos tiempos de inferencia promedian 1.52 y 1.49 segundos, respectivamente. En cuanto a precisión, YOLOv8 logra resultados impresionantes con un porcentaje superior al 90% con los entrenamientos, destacándose por su capacidad para identificar objetos correctamente en diversas condiciones, mientras reduce errores como falsos positivos. Comparado con modelos como SSD, que alcanza solo un 39.26%, y Faster R-CNN, con 67.7%, YOLOv8 demuestra una superioridad significativa. Además, las mejoras en su diseño, como la integración por defecto de funciones de pérdida avanzadas basadas en CIoU y técnicas optimizadas de entrenamiento, garantizan un rendimiento confiable en tareas de detección. Por último, YOLOv8 maximiza el uso eficiente de recursos

computacionales, con un error absoluto medio de apenas 0.213 en tareas como el conteo de personas, superando ampliamente a modelos como HOG y SSD, que presentan errores mucho mayores. Además, su arquitectura compacta y eficiente permite implementarlo en dispositivos con recursos limitados, lo que lo hace especialmente versátil para aplicaciones en sistemas embebidos y tecnologías de bajo costo [25]. Las métricas de entrenamiento y validación obtenidas a lo largo de las 100 épocas evidenciaron la estabilidad del modelo. La precisión del entrenamiento se mantuvo en un nivel alto desde las primeras iteraciones, con una ligera mejora a medida que avanza el proceso. Sin embargo, se observó un ligero estancamiento en la mejora de la precisión después de la época 75, lo que indica la necesidad de técnicas adicionales de optimización para seguir mejorando el rendimiento.

Por tal motivo, se utilizó la red neuronal de YoloV8 tomando en cuenta los recursos que disponíamos como una computadora con procesador i3 y una cámara web de resolución media. Al concluir las pruebas en la implementación podemos demostrar que se obtuvo una precisión de 85.36% en detecciones y contabilización en tiempo real (ver **Figura 32**). Esto demuestra que con solo 200 imágenes se pudo obtener buenos resultados el cual si se incrementa el dataset a unas mil imágenes y se cuenta con buenos recursos computacionales para el entrenamiento se podría obtener resultados incluso por encima del 95%.

#### 4. DISCUSIÓN

En el trabajo de Muñoz [5], Desarrollo de un algoritmo basado en python que permita entrenar una red neuronal, para conteo automático del flujo de personas que circulan, obtuvo una precisión de 80 % pero a partir de 10 personas la precisión tiende a disminuir debido al solapamiento entre ellas. En Cobos & Silva, Sistema de detección de aves mediante análisis de imágenes, obtuvo una precisión de 80% determina que las CNN presentan un buen rendimiento [7]. Al final con el trabajo de Luna & Rodríguez, Detección y conteo de personas en espacios cerrados utilizando estrategias basadas en visión artificial, obtuvo una precisión del 65% pero los beneficios no se revelaron a lo que se esperaba [26].

Este proyecto aporta un avance significativo en el campo de la inteligencia artificial aplicada al conteo de aves al superar los retos y limitaciones observados en trabajos previos. En los estudios mencionados, se logró una precisión del 80% en el conteo de personas, al igual que en la detección de aves con CNN el sistema presenta una disminución considerable en precisión al enfrentar solapamientos, lo que reduce su eficacia en escenarios complejos. La implementación desarrollada en este proyecto logró un notable avance al alcanzar una precisión del 85.36 % en la identificación y recuento de aves con un nivel de exactitud que supera de manera consistente los resultados de los trabajos anteriores, evidenciando que la inteligencia artificial se optimiza adecuadamente siendo una solución más confiable y precisa frente a métodos manuales. Además, la capacidad del sistema para mantener su desempeño en condiciones más exigentes, como solapamientos o escenarios de alta densidad, destaca su robustez y aplicabilidad. Esta tecnología ofrece una herramienta práctica y eficaz para empresas avícolas, mejorando la administración de inventarios y reduciendo errores humanos, lo que asegura una gestión más eficiente y confiable a largo plazo. También abre la posibilidad de escalar el sistema a otros animales o contextos, como la monitorización de ganado o fauna en

reservas naturales ampliando su alcance y posicionándolo como una solución adaptable para diversos sectores, desde la agricultura, avicultura y hasta la conservación ambiental.

#### 5. CONCLUSIONES

Se entrenó un conjunto de datos que incluyen 132 imágenes de aves, organizadas en categorías para entrenamiento (116 imágenes), validación (11 imágenes) y en prueba (5 imágenes), asegurando una cobertura adecuada de las variaciones en condiciones de luz, ángulos y posiciones de las aves. Esta diversidad en el dataset permitió al modelo adaptarse mejor en los entornos reales de una granja. La estructura del dataset, junto con el proceso de procesamiento en Roboflow, garantizó que el algoritmo está bien equipado para identificar y contar con precisión en contextos variados.

La implementación fue exitosa de la red neuronal YOLOv8 y Python para el coteo de aves en tiempo real en un ambiente de granja demuestra la capacidad de esta tecnología para realizar detecciones de alta precisión y velocidad mediante una cámara. Este modelo optimiza el monitoreo constante de las aves, que permite llevar un control más exhaustivo sobre el inventario en diferentes condiciones ambientales. Al mejorar la eficiencia operativa y deducir tiempos, esta herramienta permite a las empresas avícolas mantener estándares más altos de producción.

La red neuronal artificial implementada alcanzó una precisión del 85.36% en la detección y conteo de aves, lo que confirma que la inteligencia artificial es una solución efectiva frente al conteo manual tradicional, que suele ser más propenso a errores y demandas de tiempo, esta tecnología representa una alternativa confiable y eficiente que puede aplicarse en empresas avícolas, mejorando la exactitud de la contabilización y optimizando los procesos de gestión de inventarios de aves.

#### 6. REFERENCIAS

- [1] S. T. Contreras Flores, «Panorama y mercado del huevo de gallina.» 2019. [En línea]. Available: <http://surl.li/ildvkt>.
- [2] MIDAGRI, «Producción y Comercialización de Productos Avícolas,» Febrero 2023. [En línea]. Available: <http://surl.li/lollrr>.
- [3] Asociación Peruana de Avicultura, «Pollitas bebé en postura,» Julio 2022. [En línea]. Available: <http://surl.li/rmewst>.
- [4] P. Jiang, D. Ergu, F. Liu, Y. Cai y B. Ma, «Una revisión de los desarrollos del algoritmo Yolo,» 2021. [En línea]. Available: DOI: 10.1016/j.procs.2022.01.135.
- [5] L. M. Muñoz Alejandro, «Desarrollo de un algoritmo basado en python que permita entrenar una red neuronal, para conteo automático del flujo de personas que circulan,» 2023. [En línea]. Available: <https://goo.su/DUVXN>.
- [6] C. Mota-Delfin, C. Juárez-González y J. C. Olguín-Rojas, «Clasificación de manzanas utilizando visión artificial y redes neuronales artificiales,» [En línea]. Available: <https://goo.su/Gm2NK>.
- [7] L. M. Silva Labanda y J. C. Cobos Torres, «Sistema de detección de aves mediante análisis de imágenes,» 2021. [En línea]. Available: <https://goo.su/YyIm4m>.
- [8] S. K. Gutiérrez Hoyos, J. d. P. Jacinto Paredes, K. M.

- Miranda Mego, J. A. Peche Mestanza y R. D. Reyes Zavaleta, «Diseño e implementación de un sistema de control de temperatura para un galpón de pollos de la Avícola Florián e Hijos S.A.C.» 2021. [En línea]. Available: <https://goo.su/yN1R>.
- [9] M. Valle Palomino, E. Lara Carranza, L. Calderón Rodríguez y M. Iglesias Rojas, «Sistema de implementación de reconocimiento automático de asistencias mediante QR que son generados en Python para las Mype.» 2023. [En línea]. Available: <https://goo.su/RWbremR>.
- [10] A. Crespo Rodríguez, «Sistema de detección y estimación de pose de objetos basado en visión por computador para planta piloto industria 4.0.» 2019. [En línea]. Available: <http://surl.li/ezqlmj>.
- [11] J. Wang, W. Li, L. Liu y Z. Li, «Revisión de Métodos y Aplicaciones de Conteo Visual para Animales de Granja.» 2023. [En línea]. Available: DOI: 10.6041/j.issn.1000-1298.2023.S1.034.
- [12] A. F. Loaiza Quintana, D. A. Manzano Herrera y L. E. Múnera Salazar, «Sistema de Visión Artificial para Conteo de Objetos en Movimiento.» 2012. [En línea]. Available: <https://goo.su/qeJeDw>.
- [13] N. Sergei y L. Alexander, «Detección de Ciervos en Imágenes para Contar Poblaciones Mediante Métodos de Visión por Computadora.» Febrero 2024. [En línea]. Available: DOI: 10.12731/2658-6649-2024-16-2-710.
- [14] M. Castrillón, D. Hernández y J. Lorenzo, «Conteo de personas con un sensor RGBD comercial.» Septiembre 2014. [En línea]. Available: DOI: 10.1016/j.riai.2014.05.006.
- [15] L. Zhang, X. Zhou, B. Li, H. Zhang y Q. Duan, «Método de conteo automático de camarones utilizando imágenes locales y YOLOv4 liviano.» 2022. [En línea]. Available: DOI: 10.1016/j.biosystemseng.2022.05.011.
- [16] H. G. Akçay, B. Kabasakal, D. Aksu, N. Demir, M. Öz y A. Erdoğan, «Conteo automatizado de aves con aprendizaje profundo para mapeo regional de distribución de aves.» July 2020. [En línea]. Available: DOI: 10.3390/ani10071207.
- [17] L. Cao, Z. Xiao, X. Liao, Y. Yao, K. Wu, J. Mu, J. Li y H. Pu, «Conteo automatizado de pollos en ambientes de cámaras de vigilancia basado en el algoritmo de supervisión puntual: Lc-densefcn.» Junio 2021. [En línea]. Available: DOI: 10.3390/agriculture11060493.
- [18] J. Ruiz Cardón, «Detección automática de aves en las vías del AVE mediante análisis de vídeo.» 07 2020. [En línea]. Available: <https://repositorio.uam.es/handle/10486/693980>.
- [19] S. Erjie, X. Zhang, Y. Fengwei, W. Zhicheng, L. Junjie y M. Guoning, «Método de recuento clasificado de pollos de cría mixta basado en YOLOV5.» 2023. [En línea]. Available: DOI: 10.23919/CCC58697.2023.10240449.
- [20] R. Padilla, S. Lima Netto y E. AB da Silva, «Encuesta sobre métricas de rendimiento para algoritmos de detección de objetos.» 2020. [En línea]. Available: DOI: 10.1109/IWSSIP48289.2020.
- [21] G. Viera Maza, «Procesamiento de imágenes usando opencv aplicado en raspberry pi para la clasificación del cacao.» marzo 2017. [En línea]. Available: <http://surl.li/xxndyb>.
- [22] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid y S. Savarese, «Intersección generalizada sobre unión: una métrica y una pérdida para la regresión de cuadro delimitador.» 2019. [En línea]. Available: <https://goo.su/BrE7yr9>.
- [23] D. M. Brito Hoyos, Mayo 2024. [En línea]. Available: <https://ri.conicet.gov.ar/handle/11336/236742>.
- [24] J. Zhang, «Research on Target Tracking and Gesture Recognition System Combining Artificial Intelligence Algorithm and Computer Vision.» Agosto 2023. [En línea]. Available: DOI: 10.1109/ICDSI60108.2023.00033.
- [25] L. Y. Bográn Ortiz y J. J. Martínez Hernández, «Comparativa de modelos de detección de objetos y personas en espacios cerrados de acceso público.» marzo 2023. [En línea]. Available: <https://orcid.org/0000-0002-3178-8676>.
- [26] A. F. Luna Camacho y N. E. Rodríguez Menjura, «Detección y conteo de personas en espacios cerrados utilizando estrategias basadas en visión artificial.» Noviembre 2017. [En línea]. Available: <http://surl.li/zezpuv>.