

Artículo

Clasificación en tiempo real de piezas de pollo en el proceso de envasado mediante modelos de detección de objetos basados en aprendizaje profundo

Dilruba Sahin¹, Orhan Torkul^{1,*}, Merve Sisci^{2,3} , Deniz Demircioglu Diren³ , Recepçion Yilmaz⁴ 
y Alpaslan Kibar⁵

¹ Industrial Ingeniería Departamento, Sakarya Universidad, Sakarya 54050, Turquía; dilruba.sahin@ogr.sakarya.edu.tr

² Industrial Ingeniería Departamento, Kütahya Dumlupınar Universidad, Kütahya 43300, Turquía; mervesisci@sakarya.edu.tr

³ Departamento de Información Sistemas y Tecnologías, Sakarya Universidad, Sakarya 54050, Turquía; ddemircioglu@sakarya.edu.tr

⁴ Negocios Escuela, Sakarya Universidad, Sakarya 54050, Turquía; rcyilmaz@sakarya.edu.tr

⁵ Departamento de Manejo Información Sistemas, Sakarya Universidad, Sakarya 54050, Turquía; kibar@sakarya.edu.tr

* Correspondencia: torkul@sakarya.edu.tr

Resumen: La carne de pollo desempeña un papel importante en la alimentación saludable de muchas personas y representa un gran volumen de comercio global. En el sector de la carne de pollo, en algunos procesos de producción, se utilizan métodos tradicionales. Los métodos tradicionales de clasificación de piezas de pollo suelen ser manuales y requieren mucho tiempo, especialmente durante el proceso de envasado. Este estudio tuvo como objetivo identificar y clasificar las piezas de pollo para su entrada durante el proceso de envasado con la mayor precisión y velocidad posibles. Para ello, se utilizaron modelos de detección de objetos basados en aprendizaje profundo. Se desarrolló un conjunto de datos de imágenes para los modelos de clasificación mediante la recopilación de datos de imágenes de diferentes piezas de pollo, como patas, pechugas, jarretes, alas y contramuslos. Los modelos se entrenaron con las variantes del algoritmo Sólo Miras Una Vez Versión 8 (YOLOv8) y las variantes del algoritmo Real-Time Detection Transformer (RT-DETR). Posteriormente, se evaluaron y compararon en función de las métricas de precisión, recuperación, puntuación F1, precisión media promedio (mAP) y tiempo medio de inferencia por fotograma (MITF). Según los resultados obtenidos, el modelo YOLOv8s superó a los demás modelos desarrollados con otras versiones de YOLOv8 y el algoritmo RT-DETR, obteniendo valores de 0.9969, 0.9950 y 0.9807 para la puntuación F1, mAP@0,5 y mAP@0,5:0.95, respectivamente. Se ha demostrado su idoneidad para aplicaciones en tiempo real con un valor MITF de 10.3 ms/imagen.

Palabras clave: partes de pollo; aprendizaje profundo; procesamiento de imágenes; detección de objetos; reducción de desperdicios y costos; RT-DETR; YOLOv8



Editores Académicos: Honggang Chen, Yuan Li y Junyu Guo

Recibió: 27 Febrero 2025

Revisado: 21 Marzo 2025

Aceptado: 23 Marzo 2025

Publicado: 27 Marzo 2025

Citación: Sahin, D.; Torkul, O.; Sisci, M.; Diren, D.D.; Yilmaz, R.; Kibar, A. Real-Time Classification of Chicken Parts in the Packaging Process Using Object Detection Models Based on Deep Learning. *Processes* 2025, 13, 1005. <https://doi.org/10.3390/pr13041005>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. Este artículo es un artículo de acceso abierto distribuido bajo los términos y condiciones de la licencia Creative Commons Attribution (CC BY). (<https://creativecommons.org/licenses/by/4.0/>).

1. Introducción

La carne de pollo es uno de los tipos de carne más consumidos a nivel mundial debido a su alto contenido proteico y a su precio asequible. Debido al aumento del consumo de carne de pollo a nivel mundial, la necesidad de producción ha aumentado, lo que ha impulsado la expansión del sector avícola y su transformación en un mercado competitivo. La Asociación de Industriales y Criadores de Carne Blanca informó que Turquía ocupa el séptimo lugar en el comercio mundial de producción de carne de pollo [1]. Según datos del Instituto de Estadística de Turquía

[2], la producción de carne de pollo de Turquía en 2023 fue de 2,328,791 toneladas. A nivel mundial, ha aumentado de forma constante cada año, alcanzando aproximadamente 103,549,000 toneladas en 2023. También se prevé que supere los 104.2 millones para finales de 2024 [3].

En la industria avícola, donde se lleva a cabo una producción tan intensiva y se experimenta una intensa competencia a nivel mundial, es fundamental eliminar la intervención humana (operador), aumentar la velocidad del proceso y reducir los costos. Las etapas del ciclo de vida de la cadena de producción y consumo de pollo incluyen la cría de pollos, el procesamiento avícola, el sacrificio y envasado de la carne, la distribución, la venta minorista, el consumo y el final del ciclo de vida. La clasificación y el envasado son etapas importantes del proceso de producción de pollo [4]. Los procedimientos aplicados a lo largo del proceso de producción, incluido el envasado, deben garantizar que se mantenga la calidad y la seguridad de los alimentos [5].

En la mayoría de las plantas de producción de pollo, la separación de las piezas de pollo se realiza manualmente por operadores mediante inspección visual. Las piezas se separan manualmente en diferentes contenedores y luego se envían para su envasado [6]. Estos aspectos manuales del proceso ocasionan pérdida de tiempo y, por lo tanto, reducen la eficiencia. Además, son propensos a errores y no pueden garantizar la seguridad del producto. Como resultado, pueden causar numerosas desventajas, como tiempos de producción más largos, desperdicio de producto debido a errores y un aumento de las quejas de los clientes. Por lo tanto, la necesidad de sistemas automáticos en el proceso de clasificación de piezas de pollo ha alcanzado un nivel significativo.

Los avances en inteligencia artificial, aprendizaje automático y visión artificial permiten a las empresas digitalizar sus procesos de producción, minimizando la intervención humana o manual, uno de los requisitos de la Industria 4.0, y posibilitan una alta automatización de los procesos de producción. El éxito de la Industria 4.0 reside en gran medida en la integración y adaptación de estas tecnologías al proceso de fabricación existente [7]. El uso de estas tecnologías se ha generalizado cada vez más en diversas operaciones de producción y comerciales, como el control de calidad [8–14], la colaboración entre humanos y robots [15–17], el seguimiento de cordones de soldadura [18–20], el mantenimiento predictivo [21–24], la logística y el inventario [25–29], y la seguridad laboral [30–32]. Las tecnologías existentes y en desarrollo han comenzado a mostrar su impacto en las instalaciones de producción del sector avícola, donde se experimenta una producción intensiva debido a la alta demanda.

La detección de objetos, una de las tareas fundamentales de la visión artificial, implica encontrar y clasificar objetos objetivos en imágenes. En este campo, la integración de algoritmos de aprendizaje profundo ha generado avances significativos. [33]. Entre estos algoritmos, los de dos etapas, como Faster R-CNN, ofrecen una alta precisión pero funcionan más lentamente. [34]. Los algoritmos de una sola etapa como YOLO (You Only Look Once) tienen la característica de estar orientados a la velocidad. [35,36]. También han ganado popularidad en el campo de la visión por computadora porque proporcionan una alta precisión a pesar de su pequeño tamaño y pueden entrenarse en una sola GPU. [37]. You Only Look Once versión 8 (YOLOv8), una de las versiones más actualizadas y avanzadas de la serie YOLO, ofrece una opción ideal para el rendimiento en tiempo real debido a su rápido funcionamiento y alta precisión. [38]. Sin embargo, se ha informado que el algoritmo Detection Transformer (DETR) [39], que combinó la arquitectura del

transformador con la detección de objetos por primera vez, no requiere pasos intermedios complejos en la detección de objetos y proporciona buenos resultados en las implementaciones de detección de objetos [40]. El algoritmo de Transformador de Detección en Tiempo Real (RT-DETR) [41], propuesto como resultado de las mejoras realizadas a DETR, aumentó aún más la velocidad y precisión de detección. Se ha afirmado que el algoritmo ofrece resultados más satisfactorios que otros métodos en tiempo real gracias a su rendimiento superior al utilizar plataformas de alta potencia como CUDA [42].

Este estudio busca desarrollar un modelo eficaz para la detección y clasificación en tiempo real de piezas de pollo, eliminando así los problemas mencionados, automatizando el proceso de envasado en la industria avícola. Para ello, se desarrollaron siete modelos de detección de objetos utilizando cinco versiones diferentes del algoritmo YOLOv8 basado en aprendizaje profundo (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l y YOLOv8x) y dos versiones diferentes del algoritmo RT-DETR (RT-DETR-Large y RT-DETR Extra-Large) con datos de imágenes de piezas de pollo. Se compararon los rendimientos de estos modelos. Las principales contribuciones de este estudio son las siguientes:

- Los algoritmos YOLOv8 y RT-DETR se aplicaron por primera vez en el campo de la clasificación de partes de pollo.
- La eficacia de los modelos de detección se evaluó y comparó en un conjunto de datos de imágenes de clasificación de partes de pollo de cinco categorías con diferentes combinaciones, utilizando imágenes originales de partes de pollo tomadas en un ambiente experimental.
- Tras la evaluación de siete modelos de detección de objetos basados en aprendizaje profundo con alta precisión de predicción, utilizando versiones de los algoritmos YOLOv8 y RT-DETR, se demostró que el modelo YOLOv8s es más eficaz que los modelos RT-DETR y otras versiones de YOLOv8 en este conjunto de datos.

El resto de este artículo se organiza de la siguiente manera: en la segunda sección, se examinan estudios sobre automatización mediante tecnología de procesamiento de imágenes en la industria avícola. En la tercera sección, se explican los métodos empleados en este estudio. En la cuarta sección, se presentan la metodología y la aplicación propuestas. En la quinta sección, se analizan los resultados experimentales obtenidos de la implementación. Finalmente, en la sexta sección, se presentan las conclusiones.

2. Literatura

Diferentes estudios en la literatura están respaldados por la inteligencia artificial y el procesamiento de imágenes en la industria avícola. Asmara et al. [43] clasificaron los niveles de frescura de las partes del pollo como fresco, frescura media y no fresco, utilizando algoritmos de procesamiento de imágenes y aprendizaje automático basados en sus características de color y textura. En este estudio, se utilizó el algoritmo Support Vector Machine (SVM) para fines de clasificación y se comparó con los algoritmos Decision Tree y Naive Bayes. El modelo entrenado utilizando el algoritmo SVM arrojó los mejores resultados de predicción. Este modelo logró una tasa de precisión del 58/33% para imágenes obtenidas con la cámara de un teléfono inteligente, del 98% para imágenes obtenidas con una cámara web y del 79/1% para imágenes obtenidas con un microscopio digital. Putra y Prakasa [44] determinaron la frescura de la carne de pechuga de

pollo utilizando técnicas de procesamiento de imágenes por computadora y algoritmos de aprendizaje profundo. Las pechugas de pollo se dividen en dos categorías: frescas y rancias. Se utilizó un algoritmo de red neuronal convolucional (CNN) para la clasificación. El modelo desarrollado logró una tasa de precisión del 92.9% al determinar la frescura de la carne de pollo. La investigación realizada por García et al. [45] tuvo como objetivo determinar el nivel de frescura de solo la pechuga de la carne de pollo utilizando la arquitectura VGG16 de la CNN. Después de realizar varias operaciones de preprocesamiento de imágenes, como la umbralización y la transformación morfológica, el modelo entrenado logró una precisión general del 94.11% en la clasificación de la pechuga de pollo como fresca y rancia. Nyalala et al. [46] proporcionaron una estimación en tiempo real de los pesos de las carcasas de pollo en la línea de producción utilizando métodos de procesamiento de imágenes y redes neuronales artificiales (RNA). En el estudio, se desarrollaron modelos SVM lineal, SVM cuadrático, SVM cúbico, máquina de vectores de soporte de función de base radial (RBF SVM) y ANN bayesianos con fines comparativos. El modelo ANN bayesiano tuvo un mejor rendimiento que otros modelos en la estimación de los pesos de la carcasa, la cabeza y el cuello, y el SVM RBF en la estimación de los pesos del muslo, la pechuga y las alas. Este estudio presenta un ejemplo del uso en tiempo real de algoritmos de redes neuronales en la industria avícola. You et al. [47] propusieron un método de evaluación de la calidad mediante el examen de imágenes a color de carne de pollo. Las imágenes se recopilaron colocando tarjetas de color junto a ellas para la corrección de color y extraer la información de color capturada. Se utilizó la agrupación jerárquica para examinar los colores ajustados de las muestras y generar tres niveles de calidad distintos. Se ha informado que una nueva muestra de carne puede clasificarse en una de tres clases de calidad utilizando el modelo de agrupación propuesto, y su color puede utilizarse para reflejar el resultado de la evaluación de calidad. Los estudios revisados han demostrado la utilidad de la visión artificial, el aprendizaje automático y los métodos de aprendizaje profundo en diversas áreas de la industria avícola.

Cuando se examina la literatura, se puede ver que se ha logrado un gran progreso mediante la realización de investigaciones en áreas como la evaluación de la calidad de la carne de pollo, la frescura y la estimación del peso de las carcasas de pollo. Sin embargo, el número de estudios centrados en la clasificación de las partes del pollo durante el proceso de envasado y el uso de procesamiento de imágenes e implementaciones de visión artificial es limitado. En un estudio realizado por Teimouri et al. [6], se desarrolló un sistema que clasificaba las partes del pollo como pechugas, filetes, alas, patas y muslos utilizando procesamiento de imágenes y ANN. Este es un ejemplo de un sistema que puede separar las partes del pollo en tiempo real utilizando modelado inteligente basado en visión. Los aspectos geométricos, el color y las características de textura se extrajeron de los datos de imagen de las partes del pollo obtenidos utilizando una cámara de dispositivo de carga acoplada (CCD). Los modelos de clasificación se desarrollaron utilizando regresión de mínimos cuadrados parciales, análisis discriminante lineal y métodos de ANN en las características obtenidas. El modelo de ANN con un tiempo de procesamiento de 22.2 ms para una imagen mostró el mejor rendimiento con una tasa de precisión general del 93%. Salma et al. [48] clasificaron las aves de corral como pollo, pavo, avicultor y Fayoumi. Se recopilaron imágenes de las partes del muslo, el ala, el cuello y la pechuga de los animales de la especie deseada, comprados en el mercado de animales. La clasificación se realizó utilizando el

algoritmo MobileNetv2 entrenado previamente. El modelo alcanzó una precisión de hasta el 98%. En un estudio de Chen et al. [49], la detección y clasificación en tiempo real de partes de pollo en la línea de producción se lograron con el procesamiento de imágenes y el algoritmo YOLOv4. Luego, se compararon los métodos de detección YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3 y SSD-VGG16, y se concluyó que el algoritmo con el mejor rendimiento de reconocimiento en tiempo real fue YOLOV4. El modelo YOLOV4-CSPDarknet53, con un tiempo de procesamiento de 15 ms, alcanzó una tasa de precisión del 98.86%. [50] desarrolló un modelo de detección con el algoritmo Swin-Transformer para la clasificación automática de partes de pollo en la línea de producción, y su rendimiento se comparó con el de los modelos YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3 y SSD-VGG16. El modelo Swin-Transformer desarrollado produjo resultados más exitosos que otros modelos con un valor de precisión promedio media (mAP) del 97.21% y un tiempo de detección promedio de 19.02 ms. Estos estudios realizados por Teimouri et al. [6], Chen et al. [49] y Peng et al. [50] pueden verse como estudios pioneros en la implementación de la clasificación en tiempo real del procesamiento de imágenes, redes neuronales artificiales y métodos de aprendizaje profundo en las etapas de clasificación y empaquetado de partes de pollo, que consideramos que faltan en la literatura y que contribuiremos a la industria.

Aunque los modelos desarrollados en los estudios revisados mostraron un alto rendimiento, existe la necesidad de modelos con un poder predictivo más preciso y rápido. Por lo tanto, se han desarrollado versiones más actualizadas y avanzadas de estos algoritmos, como YOLOv8 y RT-DETR. El éxito de estos algoritmos, que tienen aplicaciones en diferentes campos, en comparación con otros algoritmos, ha sido probado experimentalmente. Tamang et al. [51] estudiaron un modelo de procesamiento de imágenes que detecta y clasifica situaciones de uso de mascarillas faciales, que surgió como una necesidad después de la pandemia de COVID-19 que sacudió al mundo en los últimos años. Clasificaron las situaciones de uso de mascarillas como con mascarilla puesta, sin mascarilla y con uso incorrecto. Utilizaron los algoritmos YOLOv5 y YOLOv8 para la detección y clasificación de objetos y compararon sus rendimientos. En casi todas las medidas de rendimiento, el algoritmo YOLOv8 tuvo un mejor rendimiento que el algoritmo YOLOv5. En un estudio realizado por Bawankule et al. [52], se desarrolló un sistema automático de clasificación de residuos para abordar el problema de los residuos sólidos y hacer que el proceso sea más eficiente y confiable. El modelo YOLOv8 se utilizó para detectar y clasificar residuos sólidos, como latas, baterías, desperdicios de comida, plásticos blandos, botellas y cartón. En este estudio, se compararon los rendimientos de los modelos Faster R-CNN, SSD, YOLOv4, YOLOv5 y YOLOv7, y el modelo YOLOv8 dejó atrás a los otros modelos con su alta tasa de precisión del 97.7%. El estudio de Jun et al. [53] tuvo como objetivo proporcionar detección aérea de objetos en áreas urbanas. Para lograr este objetivo, utilizaron conjuntos de datos de detección aérea de objetos en zonas urbanas con cuatro clases: personas, vehículos pequeños, vehículos medianos y vehículos grandes. Desarrollaron modelos de detección de objetos utilizando tres versiones diferentes del algoritmo YOLOv8 y dos versiones diferentes del algoritmo RT-DETR y evaluaron su rendimiento. Como resultado, se afirmó que se obtuvo un mejor modelo en comparación con los otros modelos en el estudio al obtener un mAP@0.5:0.95 de 0.598 con el modelo RT-DETR-r50 desarrollado. El modelo YOLOv8n fue el más rápido, con una velocidad de

inferencia de 30.4 cuadros por segundo (FPS). Guemas et al. [54] utilizaron el algoritmo RT-DETR para detectar especies de *Plasmodium*. Se utilizó un conjunto de datos de 24,720 imágenes recopiladas de 475 frotis sanguíneos delgados para desarrollar el modelo. El rendimiento del modelo RT-DETR se comparó con los modelos desarrollados utilizando los algoritmos YOLOv8x y YOLOv5x, y se demostró su superioridad. Debido al alto desempeño de los modelos desarrollados con los algoritmos YOLOv8 y RT-DETR en los estudios antes mencionados, en este estudio, a diferencia de los estudios revisados sobre clasificación de partes de pollo con procesamiento de imágenes, se utilizaron los algoritmos YOLOv8 y RT-DETR, que muestran un alto desempeño en aplicaciones de detección de objetos en tiempo real a partir de algoritmos de aprendizaje.

La literatura muestra que el número de estudios realizados con algoritmos de procesamiento de imágenes y aprendizaje profundo en el proceso de envasado en la industria avícola es limitado. Además, hasta donde sabemos, no se ha encontrado ninguna implementación de los algoritmos YOLOv8 o RT-DETR en esta área. Al aprovechar estos algoritmos avanzados, que han demostrado un rendimiento de clasificación superior en otros campos, esta investigación llena un vacío crítico en la literatura y proporciona un enfoque novedoso para automatizar la clasificación de piezas de pollo en tiempo real.

3. Métodos

Este estudio, cuyo objetivo era la clasificación automática de partes de pollo, requería un modelo de detección de objetos rápido y eficaz para el reconocimiento de objetos en tiempo real. Por lo tanto, se utilizaron dos versiones diferentes de modelos basados en aprendizaje profundo: YOLOv8 y RT-DETR.

3-1. Sólo miras una vez versión 8 (YOLOv8)

El algoritmo YOLOv8 (You Only Look Once versión 8) [55], desarrollado por Ultralytics para lograr un mejor rendimiento e incluye las características de alto rendimiento de las versiones anteriores de YOLO, también incluye una nueva red troncal, un cabezal de detección sin anclas y una nueva función de pérdida, además de otras versiones de YOLO [56]. Este algoritmo, que utiliza una sola CNN y está configurado para detectar y segmentar objetos en imágenes, estima cuadros delimitadores y probabilidades de clase [57]. En el modelo YOLOv8, a diferencia del modelo YOLOv5 (You Only Look Once versión 5), en lugar de utilizar anclas para medir la distancia de un objeto, el centro del objeto se estima con detección sin anclas [58]. La estructura de red del algoritmo, que consta de las secciones de capa de entrada, troncal, cuello y cabeza, se muestra en la Figura 1.

La imagen alimentada a la capa de entrada se escala a un tamaño fijo y se realizan operaciones de preprocesamiento. Posteriormente, se envía a la red troncal CSPDarknet53, también utilizada en YOLOv7, para extraer características. En este caso, las características se extraen de las imágenes mediante capas convolucionales [59]. La sección de la red troncal contiene el CSPBottleneck con dos bloques de convolución (C2F) y un bloque de agrupación rápida de pirámides espaciales (SPPF). El módulo C2F utilizado para la extracción de características mejora los resultados de detección al obtener información más completa sobre el flujo de gradiente. También reduce la red para aligerar el modelo. El módulo SPPF, ubicado al final de la red troncal, reduce el número de capas de la red mediante la capa de agrupación máxima para eliminar operaciones innecesarias. La parte del cuello, que realiza la agregación multiescala, contiene una red de agregación de rutas (PAN) y una red de pirámides de características (FPN). Esta parte combina las características de bajo y alto nivel que le transmite la red del cuello y luego las

envía a la parte de la cabeza [41, 60].

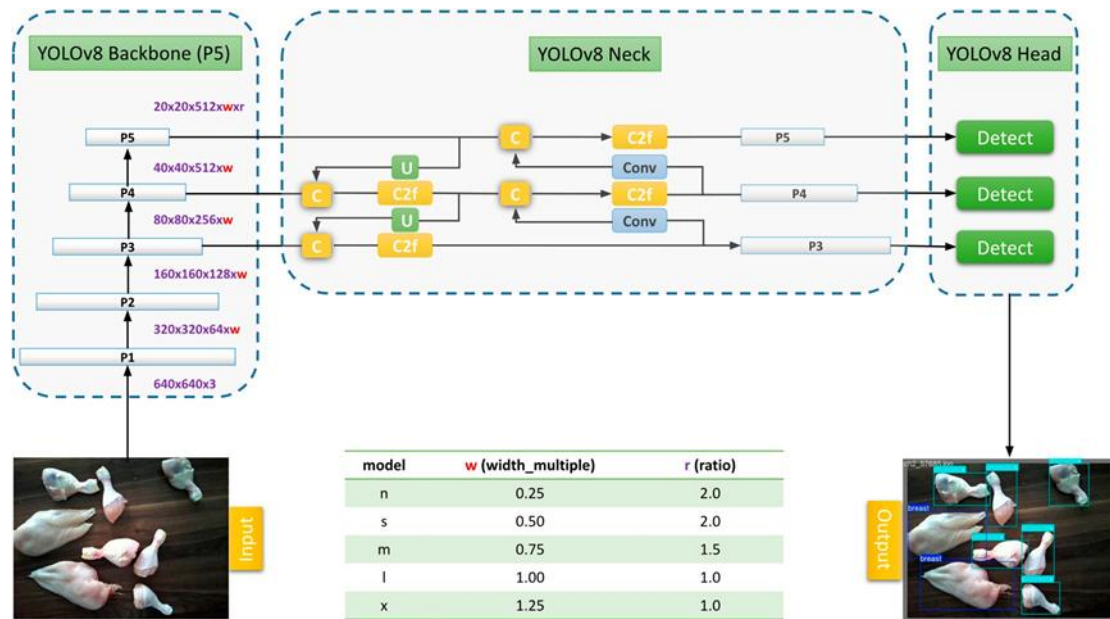


Figura 1. La estructura de YOLOv8.

La parte principal del algoritmo consta de tres capas básicas: clasificación, regresión del cuadro delimitador y puntuación de confianza. La capa de clasificación procesa las características de la capa FPN para determinar la clase de objetos, la capa de regresión del cuadro delimitador para estimar las coordenadas de los cuadros delimitadores de los objetos en la imagen y la capa de puntuación de confianza para estimar si los objetos aparecerán en la imagen [61]. Se utilizan dos tipos de funciones de activación en el algoritmo YOLOv8: softmax y sigmoide. La función softmax clasifica el objeto en la imagen, mientras que la función sigmoide determina si el objeto está dentro del cuadro delimitador. Además, el uso de funciones de pérdida centrada en la distribución (DFL) y la intersección completa sobre unión (CIoU) en el algoritmo aumenta la precisión de detección para objetos más pequeños [62]. La infraestructura de Ultralytics permite cinco tipos de implementaciones del algoritmo YOLOv8 que difieren en varias características, como el número de capas y parámetros. YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l y YOLOv8x representan los modelos nano, pequeño, mediano, grande y extragrande de YOLOv8, respectivamente. En este caso, el modelo YOLOv8n presentó el menor número de parámetros.

3-2. Transformador de detección en tiempo real (RT-DETR)

Desarrollado para aplicaciones de detección de objetos en tiempo real, el algoritmo Real-Time Detection Transformer (RT-DETR) [63] fue lanzado por el equipo Paddle de Baidu en Enero de 2023 como una tecnología de detección de objetos de extremo a extremo que integra arquitecturas de transformadores y CNN. El algoritmo, que puede aumentar significativamente la velocidad de procesamiento manteniendo una alta precisión, se desarrolló con base en el modelo DETR [64, 65]. Originalmente desarrollado para tareas de procesamiento de lenguaje natural, el modelo DETR, que utiliza transformadores para codificar y decodificar para predecir directamente los cuadros delimitadores y las clases de objetos, fue desarrollado por Facebook AI Research en 2020. Esto simplifica el proceso de detección de objetos al eliminar los pasos de posprocesamiento y las propuestas de región tradicionales. Realiza perfectamente la tarea de detección de

múltiples objetos, incluso en imágenes con fondos complejos [66]. En el enfoque DETR, que no requiere anclas predefinidas o cuadros candidatos, el problema de detección de objetos se transforma en un problema de consulta de objetos. Durante la operación del algoritmo, las posiciones de los píxeles se convierten en un vector de consulta en forma de una serie de vectores de características mediante el codificador transformador. El decodificador utiliza estos vectores de consulta para predecir la clase de objetos mediante interacciones con sus cuadros delimitadores [64]. Sin embargo, para superar el alto coste computacional de DETR, que supone un reto significativo para las aplicaciones en tiempo real, se desarrolló el modelo RT-DETR [67].

El modelo RT-DETR elimina el componente de supresión no máxima (NMS) para resolver el problema de latencia en los modelos existentes de detección de objetos en tiempo real [68]. Las mejoras clave que proporcionan un rendimiento superior en términos de velocidad y precisión para aplicaciones en tiempo real incluyen un codificador híbrido que combina CNN y transformadores para gestionar eficientemente las características multiescala separando las interacciones intraescala y la fusión interescala, y una selección mínima de consultas que mejora la inicialización de las consultas del codificador/decodificador de objetos [65, 69, 70]. Los componentes del algoritmo RT-DETR se muestran en la Figura 2. Estos son un decodificador de transformador (con un encabezado de predicción auxiliar), un codificador híbrido y una red troncal.

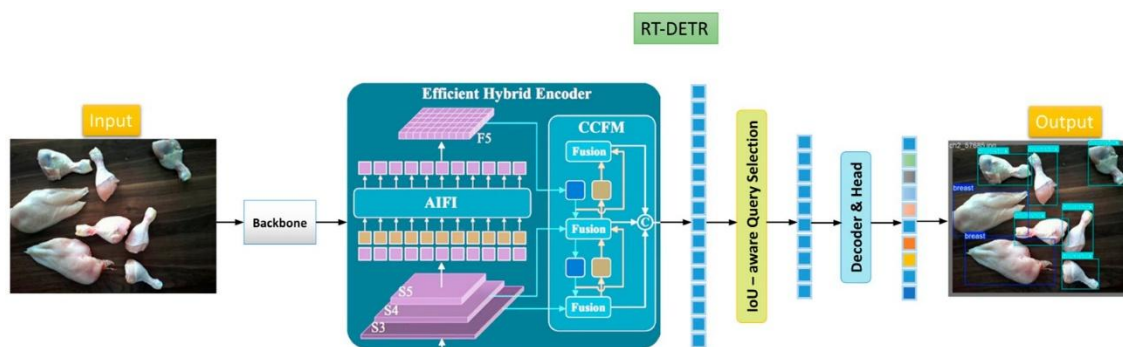


Figura 2. La arquitectura de RT-DETR.

Con la imagen como entrada, la red troncal, que constituye la primera capa de procesamiento, utiliza estructuras CNN como la serie ResNet o HGNet para extraer características básicas a diferentes niveles, como características de nivel alto, medio y bajo, con un muestreo reducido de 32, 16 y 8 veces, respectivamente [64,68]. Los resultados se extraen de la parte de la red troncal del algoritmo en tres escalas: S3, S4 y S5 [70].

El codificador híbrido, que convierte las características multiescala de la red troncal en una secuencia de características de imagen, cuenta con dos mecanismos básicos: el módulo de Fusión de Características entre Escalas (CCFM) basado en CNN y el módulo de Interacción de Características Intraescala Basada en la Atención (AIFI) [71]. El módulo AIFI se aplica para extraer características de alto nivel más avanzadas. El módulo CCFM, que utiliza varias capas convolucionales, se aplica para combinar características a diferentes niveles [68]. El decodificador de transformador, que recibe las características multiescala procesadas del codificador híbrido, se encarga de refinar estas características para generar los resultados finales de detección de objetos. Estos resultados consisten en cuadros delimitadores y las puntuaciones de clasificación de los objetos en la imagen. Este proceso se realiza mediante una serie de capas del decodificador de transformador que refinan

iterativamente las consultas de objetos. El encabezado de predicción auxiliar que contiene produce resultados de detección intermedios, a la vez que garantiza que las predicciones que generará el modelo se mejoren a lo largo de las etapas [67].

4. Metodología e Implementación

La metodología propuesta en este estudio para el desarrollo de modelos de clasificación de partes de pollo se ilustra en la Figura 3. Tras obtener las imágenes de las partes de pollo en el ambiente experimental, se aplicaron dos procesos de preprocesamiento de datos: etiquetado y división de datos, para asegurar que el conjunto de datos estuviera en un formato compatible con los modelos. Posteriormente, se desarrollaron siete modelos de detección de partes de pollo basados en los algoritmos YOLOv8 y RT-DETR, y se evaluó su rendimiento.

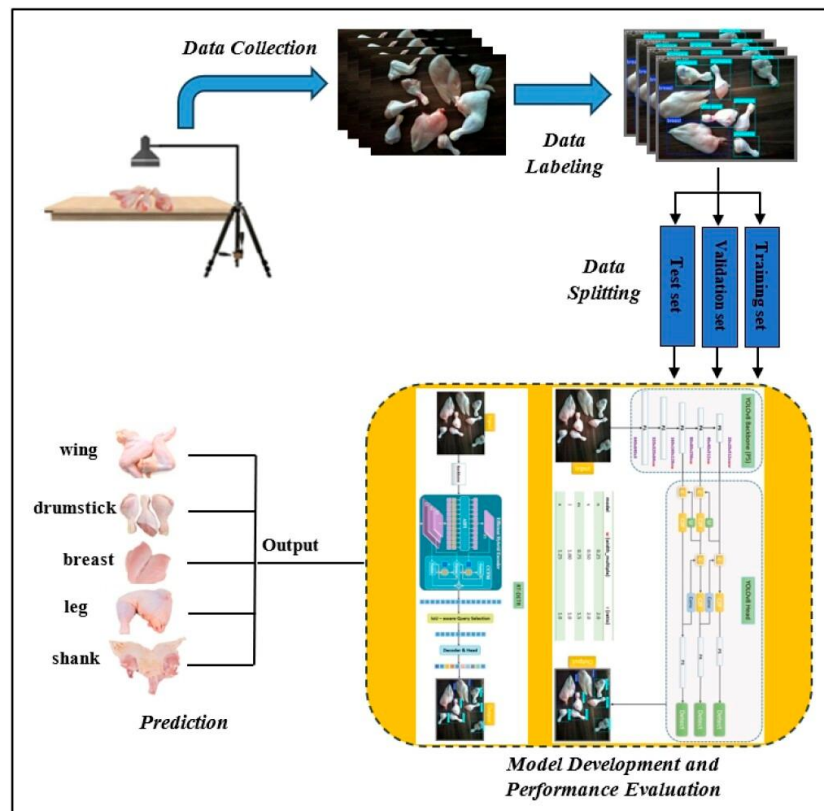


Figura 3. Los pasos de la metodología del estudio.

4-1. Conjunto de datos de identificación de partes de pollo

Para que un modelo de aprendizaje profundo sea exitoso, este debe entrenarse con un conjunto de datos suficiente que contenga una cantidad suficiente de ejemplos por clase y que pueda representar todo el espacio de datos [72]. Por lo tanto, en este estudio, se recopilaban imágenes de cinco tipos de piezas de pollo: pierna, pechuga, jarrete, ala y contramuslo, y se generó un conjunto de datos de imágenes para la clasificación de piezas de pollo. Para ello, se adquirieron productos envasados de cada tipo de pollo en la carnicería de un supermercado, en cantidad suficiente para cada tipo, teniendo en cuenta las fechas de producción y la frescura de los productos. Las piezas de pollo resultantes se colocaron en diferentes combinaciones de variedades, en diversas posiciones y ángulos, en el área experimental de clasificación y envasado. La Figura 4 muestra un diagrama del mecanismo utilizado para obtener imágenes de las partes del pollo. El módulo de cámara USB Arducam de 8 MP y 1080P, utilizado para la captura de imágenes, se ajustó a una posición adecuada para mantener las piezas de pollo en el plano focal

y en el centro de la vista. Este módulo se conectó a una computadora portátil con un procesador Intel® Core™ i7-7700HQ de 2.80 GHz y 16 GB de RAM. Con la ayuda de la cámara, la computadora portátil y los códigos desarrollados con el lenguaje de programación Python, se capturaron imágenes de las partes del pollo con una resolución de 640×480 a 30 fps y se guardaron en una computadora portátil. La biblioteca OpenCV, ampliamente utilizada en aplicaciones de procesamiento de imágenes, se utilizó para desarrollar los códigos.

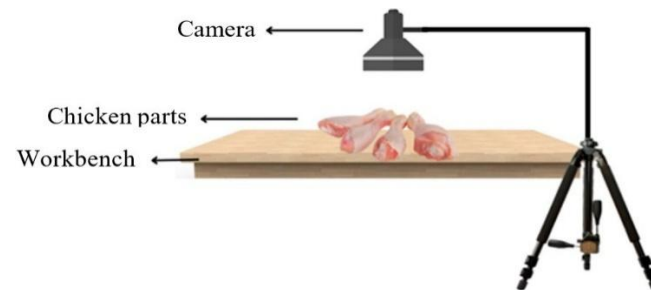


Figura 4. Una imagen representativa del mecanismo de adquisición de imágenes.

Las imágenes de pollos se obtuvieron a la luz del día con la parte superior abierta para aprovechar la flexibilidad de las condiciones ambientales, como el movimiento y la luz. Se obtuvieron 907 imágenes de todas las variedades colocando las piezas en diferentes posiciones sobre la mesa, girándolas de diversas maneras y recopilando sus imágenes. Las imágenes obtenidas de las piezas de pollo se muestran en la Figura 5. El conjunto de datos completo incluyó 547 pechugas, 1028 muslos, 464 alas, 209 patas y 154 chambaretes.

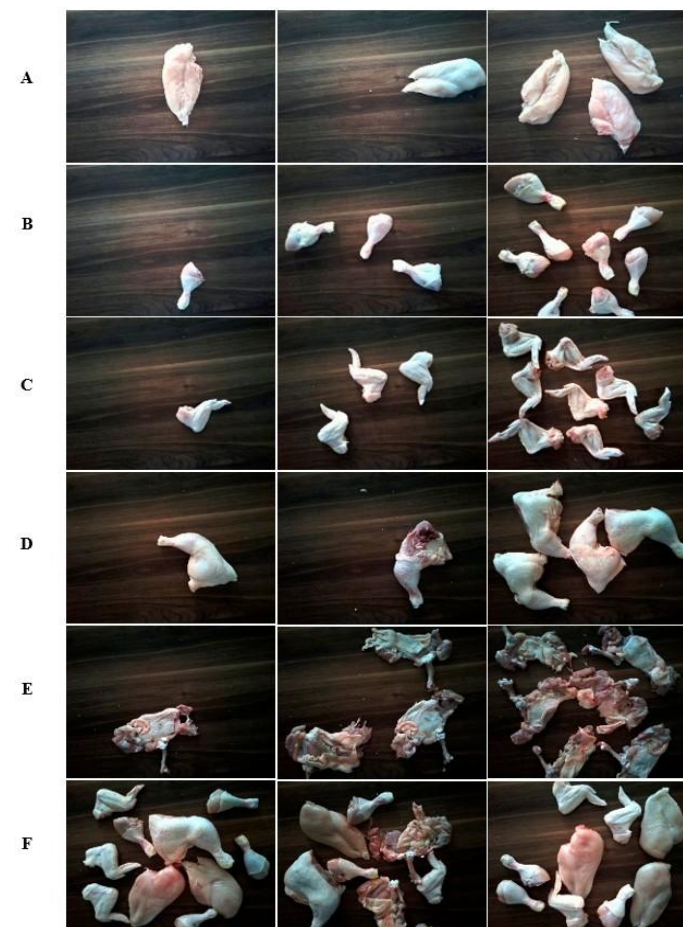


Figura 5. Ejemplos de imágenes obtenidas de las siguientes variedades de partes de pollo: (A) pechuga, (B) muslo, (C) ala, (D) pierna, (E) jarrete y (F) variedades de partes de pollo enteras.

4-2. Preparación y preprocesamiento de datos

La preparación y el preprocesamiento de los datos se realizaron en dos pasos. Primero, se etiquetaron las imágenes con el software LabelImg 1.8.6, una herramienta de etiquetado de datos de código abierto basada en Python. Posteriormente, el 70% de las imágenes etiquetadas se asignaron aleatoriamente para el entrenamiento de los modelos, el 15% para la validación y el 15% para la prueba de los modelos entrenados. El número de imágenes y piezas de pollo en los conjuntos de datos se presenta en la Tabla 1.

Tabla 1. La cantidad de imágenes y partes de pollo en los conjuntos de datos como resultado de la división de datos.

Clase	Conjunto de datos de entrenamiento		Conjunto de datos de validación		Conjunto de datos de prueba	
	Imágenes	Instancias	Imágenes	Instancias	Imágenes	Instancias
Todo	634	1639	136	400	137	363
pechuga	261	384	51	73	58	90
muslo	303	696	74	198	61	134
ala	106	315	21	63	24	86
patas	112	146	30	40	17	23
muslos	38	98	9	26	9	30

4-3. Métricas de evaluación del desempeño

Siete modelos diferentes de detección de partes de pollo, basados en YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l y YOLOv8x) y RT-DETR (RT-DETR Large y RT-DETR Extra-Large), entrenados con imágenes de partes de pollo, se probaron con los datos de imagen asignados para las pruebas durante la fase de preparación de datos. Para evaluar los resultados de las pruebas, se utilizaron métricas de evaluación del rendimiento de varios modelos de detección de objetos, incluyendo precisión, recuperación, puntuación F1, mAP@0.5, mAP@0.5:0.95 y tiempo medio de inferencia por fotograma (MITF).

La métrica de rendimiento de precisión, que mide la precisión de detección del modelo, mide el porcentaje de piezas de pollo detectadas correctamente del número total de piezas detectadas. La métrica de rendimiento de recuperación, que mide el número de piezas de pollo que el modelo omite accidentalmente, se calcula como la relación entre las piezas de pollo detectadas correctamente y el número total de piezas de pollo detectadas correctamente y omitidas. Esto es importante en aplicaciones donde no se deben producir errores, como la detección de piezas de pollo [73]. La puntuación F1, que se calcula como la media armónica de los valores de precisión y recuperación, se utiliza para medir el equilibrio entre ambas métricas de rendimiento [74].

Las métricas de rendimiento de precisión, recuperación y puntuación F1 se pueden calcular como en las ecuaciones (1)–(3) [75], donde CN_{TP} es el número de piezas de pollo detectadas correctamente por el modelo, CN_{TN} es el número de piezas de pollo omitidas correctamente y CN_{FP} es el número de piezas de pollo detectadas erróneamente.

$$Precision = \frac{CN_{TP}}{CN_{TP} + CN_{FP}} \quad (1)$$

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

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (3)$$

Otra métrica importante de rendimiento en los modelos de detección de objetos es la precisión media promedio (mAP). Esta métrica expresa la precisión promedio del modelo para diferentes clasificaciones. El valor de la precisión media (AP) para una sola clase se obtiene calculando el área bajo la curva de precisión-recuperación (PR), como se expresa en la ecuación (4) [70].

$$AP = \int_0^1 Precision(Recall) d(Recall) \quad (4)$$

El criterio de rendimiento mAP se determina calculando los valores promedio de AP para todas las clases, como se muestra en la Ecuación (5). Aquí, k representa la clase de pieza de pollo relevante y N representa el número total de clases.

$$mAP = \frac{1}{N} \sum_{k=1}^N AP_k \quad (5)$$

La intersección-unión (IOU), un cálculo de razón, se utiliza en los cálculos de mAP y se obtiene dividiendo la intersección del área real del cuadro delimitador entre el área total [76]. Cuando este valor se acepta como valor umbral, se pueden calcular los valores del criterio de rendimiento de mAP@0.5 y mAP@0.5:0.95. La tasa de objetos detectados con una precisión del 50% (mAP@0.5) se obtiene cuando el valor de IOU se establece en 0.5. Por el contrario, la tasa de objetos detectados con una precisión del 50-95% se obtiene al asignar múltiples valores para el valor de IOU entre 0.5 y 0.95, aumentando en intervalos de 0.05.

El criterio MITF utilizado en este estudio expresa el tiempo de inferencia que el modelo requiere para detectar en una sola imagen [77].

5. Experimentos y resultados

5-1. Diseño experimental

Se utilizó el lenguaje de programación Python en las fases de entrenamiento y prueba de todos los modelos basados en los algoritmos YOLOv8 y RT-DETR. Los experimentos se realizaron en un ambiente de desarrollo con una combinación de la infraestructura de aprendizaje profundo TensorFlow, Python 3.12.4, Pytorch 2.3.1, Ultralytics 8.3.96, CUDA12.5 y CUDNN9.2 en un sistema operativo Windows 10. El hardware utilizado para los experimentos es una estación de trabajo con un procesador Intel® Core™ i7-12700H de 12ª generación a 2.30 GHz, 16 GB de RAM y una tarjeta gráfica NVIDIA GeForce RTX 3060.

La Tabla 2 presenta los valores de los hiperparámetros utilizados para entrenar los modelos. Todas las imágenes del conjunto de datos se redimensionaron a 640×640 píxeles, y la red se alimentó con estas imágenes durante los procesos de entrenamiento y prueba. Diferentes valores de parámetros establecidos durante el proceso de entrenamiento de modelos basados en aprendizaje profundo como YOLOv8 y RT-DETR conducen a diferentes resultados. Uno de estos parámetros, el tamaño del lote, es un parámetro que afecta el tiempo de entrenamiento y el rendimiento [56]. A medida que aumenta el tamaño del lote, disminuye el tiempo de entrenamiento. En este estudio, se probaron diferentes tamaños de lote de 2, 4, 8, 16 y 32. Se encontró que el tamaño de lote que da los

mejores valores en términos de criterios de rendimiento es 2. El tamaño de la época se estableció en 300. Se aplicó una estrategia de calentamiento en el entrenamiento. Esta estrategia mejora el rendimiento del modelo y aumenta la velocidad de convergencia [78]. En la estrategia de calentamiento aplicada, la tasa de aprendizaje se estableció para aumentar gradualmente durante tres épocas. El número de épocas para la paciencia se estableció en 100. En otras palabras, si no se observaron mejoras en las últimas 100 épocas durante el entrenamiento de los modelos, el entrenamiento del modelo se detendría. El optimizador se configuró automáticamente. El algoritmo optimizador, determinado automáticamente por la biblioteca para el entrenamiento, fue el algoritmo de optimización Adam [79]. La tasa de aprendizaje inicial y el momento determinados por el optimizador fueron 0.001111 y 0.9, respectivamente. Se aplicó la técnica de aumento de mosaico al conjunto de datos de entrenamiento. En esta técnica, se extraen aleatoriamente cuatro imágenes del conjunto de datos de entrenamiento. Posteriormente, se realiza un recorte aleatorio de estas imágenes y se combinan en una única imagen de mosaico [80].

Tabla 2. Parámetros y valores de entrenamiento para modelos de detección de objetos.

Parámetro	Valor
Época	300
Tamaño del lote	2
Impulso	0.9
Tamaño de la imagen de entrada	640 × 640
Épocas de calentamiento	3
Impulso de calentamiento	0.8
Tasa de aprendizaje del sesgo de calentamiento	0.1
Paciencia	100
Trabajadores	8
Optimizador	Adam
Disminución del peso	0.0005
Tasa de aprendizaje inicial	0.001111
Tasa final de aprendizaje	0.01

Las características de diseño de los modelos de detección de objetos basados en aprendizaje profundo desarrollados en este estudio, como el número de parámetros, el número de capas, las giga operaciones de punto flotante por segundo (GFLOP), el número de épocas completadas, el tiempo de entrenamiento y la información comparativa sobre el proceso de entrenamiento, se dan en la Tabla 3. El número máximo de épocas de 300 épocas se completó para todos los modelos YOLOv8. Debido a que no hubo mejora durante las últimas 100 épocas, el entrenamiento de ambos modelos RT-DETR se detuvo antes de tiempo. El entrenamiento del modelo RT-DETR grande se completó en 244 épocas, y el entrenamiento del modelo RT-DETR extra grande se completó en la época 158. Como se muestra en la tabla, el tiempo de entrenamiento por época aumentó a medida que aumentó el número de parámetros y capas en los modelos YOLOv8 y RT-DETR. El tiempo de entrenamiento por época en ambos modelos RT-DETR es mayor que el de todos los modelos YOLOv8. El tiempo de entrenamiento

más bajo por época se observó en el modelo YOLOv8n, mientras que el tiempo de entrenamiento más alto se observó en el modelo RT-DETR Extra grande.

Tabla 3. Detalles de entrenamiento de modelos.

Modelo	Parámetros	Número de épocas completadas		Tiempo de formación (horas)	Tiempo de entrenamiento por época (horas)	GFLOPs
YOLOv8n	3,006,623	168	300	2.244	0.00748	8.1
YOLOv8s	11,127,519	168	300	4.380	0.0146	28.4
YOLOv8m	25,842,655	218	300	5.469	0.01823	78.7
YOLOv8l	43,610,463	268	300	6.494	0.021647	164.8
YOLOv8x	68,128,283	268	300	8.693	0.028977	257.4
RT-DETR Grande	31,994,015	494	244	8.400	0.034426	103.5
RT-DETR Extra grande	65,477,711	638	158	11.176	0.070734	222.5

5-2. Resultados experimentales

Para realizar una evaluación comparativa de los modelos entrenados con los algoritmos YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x, RT-DETR Large y RT-DETR Extra-Large en términos de precisión y eficiencia de detección, los modelos se probaron en el conjunto de datos de prueba. Los valores de precisión, recall, mAP@0.5, mAP@0.5:0.95 y MITF obtenidos de los experimentos se enumeran en la Tabla 4. Los valores MITF consisten en el tiempo de inferencia que los modelos dedican al procesar una imagen en la fase de prueba. Al examinar la tabla, se puede ver que todos los modelos proporcionan valores superiores a 0.95 para precisión, recordar, Puntuación F1, mAP@0.5 y mAP@0.5:0.95. El valor de precisión más alto obtenido fue 0.9981 por el modelo RT-DETR Extra grande, seguido por el modelo YOLOv8s con un valor de 0.9961. El valor de recall más alto (1.0) se observó en los modelos YOLOv8n y RT-DETR Grande. El Puntuación F1 más alto, mAP@0.5, y mAP@0.5:0.95, se logró con el modelo YOLOv8s, con valores de 0.9969, 0.9950, y 0.9807, respectivamente. El valor de precisión más bajo de 0.9801 se obtuvo para YOLOv8n. Los valores más bajos de Puntuación F1, recall, mAP@0.5, y mAP@0.5:0.95 se lograron con el modelo RT-DETR Extra-Large con valores de 0.9835, 0.9694, 0.9888, y 0.9588, respectivamente. Se observó que a medida que aumentaba el número de parámetros de los modelos, los valores MITF generalmente aumentaban. Se concluyó que el modelo más rápido fue el YOLOv8n, con un valor de 6.1 ms/imagen, seguido del YOLOv8s, con un valor de 10.3 ms/imagen. El modelo con la velocidad de inferencia más lenta fue el RT-DETR Extra grande, con una velocidad de 45.8 ms/imagen. Considerando tanto la velocidad como otros criterios de rendimiento, se puede afirmar que el modelo YOLOv8s ofrece el mejor rendimiento.

Tabls 4. Comparación de los valores de las métricas de rendimiento de los modelos YOLOv8 y RT-DETR.

Modelo	Precisión	Recordar	Puntuación F1	mAP@0.5	mAP@0.5:0.95	MITF (ms/Imagen)
YOLOv8n	0.9801	1.0	0.9900	0.9948	0.9720	6.1
YOLOv8s	0.9961	0.9977	0.9969	0.9950	0.9807	10.3
YOLOv8m	0.9864	0.9911	0.9888	0.9926	0.9802	15.2
YOLOv8l	0.9949	0.9925	0.9937	0.9945	0.9738	22.7
YOLOv8x	0.9884	0.9910	0.9897	0.9937	0.9742	32.8
RT-DETR Grande	0.9908	1.0	0.9954	0.9938	0.9670	30.5
RT-DETR Extra grande	0.9981	0.9694	0.9835	0.9888	0.9588	45.8

Los gráficos de la curva de precisión-recuperación, la curva de confianza F1 y la curva de recuperación-confianza del modelo YOLOv8s, que proporcionó los resultados más exitosos, y el modelo RT-DETR Extra-Large, que mostró el menor éxito, se muestran en la Figura 6. La curva de precisión-recuperación (PR) es una curva donde los valores de recuperación están en el eje x, los valores de precisión están en el eje y, y el área bajo esta corresponde al valor mAP. El gran tamaño de esta área (el alto valor mAP) indica que el modelo de detección de objetos propuesto es exitoso [81]. Como se puede ver en la figura, el área bajo la curva de precisión-recuperación (PR) del modelo YOLOv8s (Figura 6d) es mayor que el área bajo la curva de precisión-recuperación (PR) del modelo RT-DETR Extra-Large (Figura 6b). Para todas las clases, el área bajo la curva característica operativa del receptor (AUC) fue de 0.989 para el modelo RT-DETR Extra grande y de 0.995 para el modelo YOLOv8s. La curva de confianza F1 puede proporcionar una puntuación de confianza visual que optimiza la recuperación y la precisión. La puntuación de confianza debe ser lo más alta posible [38]. Según los resultados de la prueba, el valor óptimo de la confianza de la medida F fue de 0.831 para el modelo YOLOv8s y de 0.586 para el modelo RT-DETR Extra-Large. Mientras que el valor máximo de la puntuación F1 del modelo YOLOv8s fue de 1.00, el valor máximo de la puntuación F1 del modelo RT-DETR Extra grande fue de 0.98. En este caso, el modelo YOLOv8s mostró una mejora de 0.02. En la Figura 7 se muestran ejemplos de los resultados de precisión de los datos de prueba del modelo YOLOv8s desarrollado en este estudio. Se observa un rendimiento muy alto tanto en la estimación de clase como en la estimación del cuadro delimitador, incluso en la detección de partes superpuestas.

Para evaluar con mayor precisión la superioridad del modelo YOLOv8s, con la mayor capacidad de detección en la detección y clasificación de partes de pollo, se evaluó su rendimiento comparándolo con otros estudios existentes sobre la clasificación de partes de pollo. La Tabla 5 presenta un resumen comparativo. Al examinar la tabla, se observa que el modelo YOLOv8s desarrollado en este estudio presenta el mejor rendimiento entre los demás estudios en términos de precisión y velocidad en la clasificación y detección de partes de pollo. Este análisis comparativo confirmó la superioridad del modelo YOLOv8s en la detección eficaz de

partes de pollo en comparación con los modelos desarrollados en otros estudios.

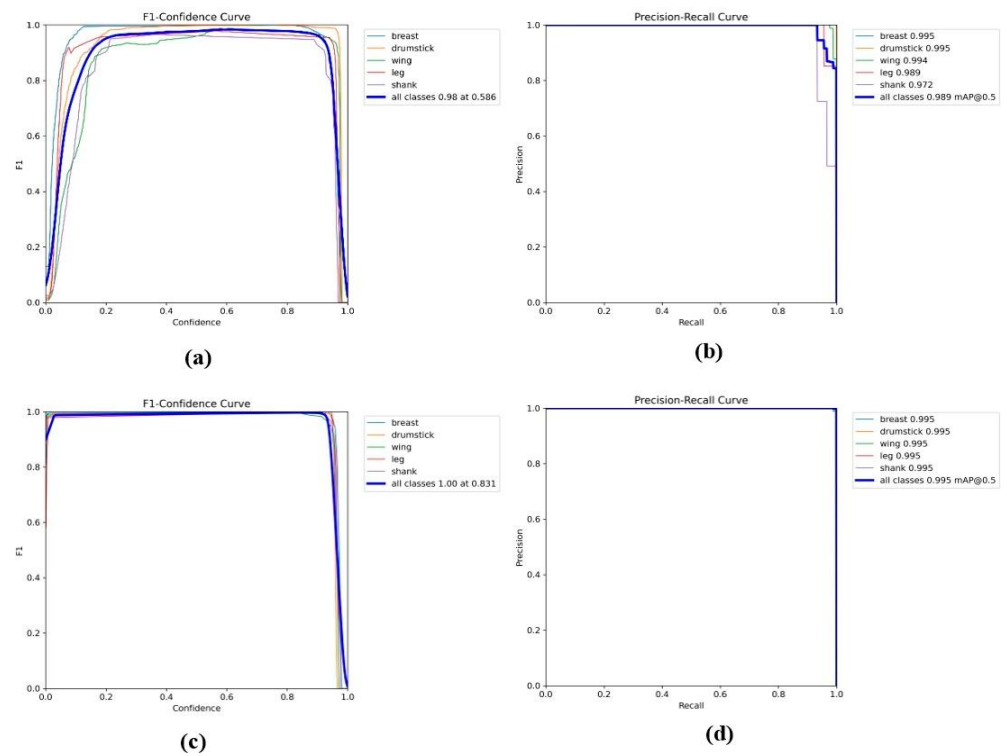


Figura 6. Curvas de precisión-recuperación y F1-confianza (a) Curva de confianza F1 del modelo RT-DETRX, (b) curva de precisión-recuperación del modelo RT-DETRX, (c) Curva de confianza F1 del modelo YOLOv8s, (d) curva de precisión-recuperación del modelo YOLOv8s.



Figura 7. Ejemplos de los resultados de precisión en los datos de prueba para el modelo YOLOv8 entrenado.

Tabla 5. Comparación entre la metodología propuesta y los estudios existentes en la literatura.

Studio	Clases (Tipos de partes del pollo)	Detalles del conjunto de datos	Métodos	El método más exitoso	Los resultados más exitosos
Teimouri et al. [6]	pechuga, pierna, filete, ala y muslo	100 muestras	Regresión de mínimos cuadrados parciales, análisis discriminante lineal y red neuronal artificial YOLOV4-CSPDarknet53, YOLOV3-Darknet53, YOLOV3-MobileNetv3, SSD-MobileNetv3, and SSD-VGG16	Artificial neuronal rojo	Precisión general: %93, MITF: 15 ms/imagen
Chen et al. [49]	wing, leg, and breast	600 imágenes	Transformador Swin, YOLOV3-Darknet YOLOV3-MobileNetv3, SSD-MobileNetv3, and SSD-VGG16	YOLOV4 CSPDarknet53-	mAP%: 98.86%, MITF 22.2 ms/imagen
Peng et al. [50]	breast	2000 imágenes	YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l and YOLOv8x) RT-DETR (RT-DETR Grande and RT-DETR Extra grande)	Transformador Swin	mAP% = 97.21%, MITF: 19.02 ms/imagen
Este estudio	muslo, ala, pierna y codillo	907 imágenes		YOLOv8s	Puntuación F1: 0.9969, mAP@0.5: 0.9950, mAP@0.5:0.95: 0.9807, MITF: 10.3 ms/imagen

6. Conclusiones

La separación manual de las piezas de pollo por parte del operador durante el proceso de envasado en las plantas de producción avícola puede provocar errores humanos, operaciones innecesarias y repetitivas, desperdicio de alimentos e incluso la insatisfacción del cliente por productos defectuosos. Por lo tanto, es necesario desarrollar un sistema automatizado para separar las piezas de pollo de forma rápida y precisa durante este proceso. Este estudio busca desarrollar un modelo de detección automática para su uso en el proceso de envasado de una planta básica de producción avícola. El sistema que detectará las piezas de pollo también las clasificará. Para lograr este objetivo, se utilizaron métodos de visión artificial basados en aprendizaje profundo. Antes de comenzar el desarrollo de los modelos, se analizó en detalle el proceso de envasado de pollo de una empresa del sector avícola en Turquía. Primero, se recopilaron imágenes de las piezas de pechuga, muslo, ala, pierna y jarrete incluidas en el proceso de envasado de la empresa en la configuración experimental preparada. El conjunto de datos constaba de 907 imágenes; para simular escenarios reales encontrados en la línea de envasado de la empresa, se proporcionaron variaciones en las condiciones de iluminación, ángulos y ligeros cambios en la posición de las

piezas de pollo. Para aumentar la robustez, se recopilaban imágenes de diferentes pollos, lo que permitió la variabilidad natural en tamaño, forma y textura. Además, se realizaron operaciones de preprocesamiento, incluyendo técnicas de redimensionamiento y aumento de mosaico, para mejorar la capacidad de generalización del conjunto de datos. La variabilidad del conjunto de datos es crucial para entrenar un modelo que se adapte eficazmente a diferentes ambientes de producción y mantenga una alta precisión en diversas condiciones. Posteriormente, se desarrollaron modelos de detección de objetos basados en aprendizaje profundo utilizando diferentes versiones de los algoritmos YOLOv8 y RT-DETR (YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x, RT-DETR Grande y RT-DETR Extra grande), y se investigó su rendimiento. Es necesario desarrollar modelos con alta precisión y velocidad en tiempo real para la clasificación de partes de pollo en el proceso de envasado. En este estudio, se seleccionaron los modelos YOLOv8 y RT-DETR, modelos de detección de objetos de vanguardia basados en aprendizaje profundo, debido a su éxito en estudios realizados en diferentes campos [51–54] y para cubrir la necesidad de clasificar partes de pollo durante el proceso de envasado. La aplicabilidad de los modelos desarrollados en un ambiente industrial se evaluó considerando su capacidad de trabajo en tiempo real y su alta precisión. Si bien esto se ha destacado en estudios realizados en diferentes campos, como los de Jin y Zhang [82], Yu y Chen [71] y Wu et al. [83] El modelo de detección de objetos RT-DETR obtuvo un mejor rendimiento que el modelo YOLOv8 en términos de precisión y velocidad. Al comparar la precisión del modelo YOLOv8s desarrollado en este estudio, se obtuvieron resultados más precisos que los modelos RT-DETR desarrollados en este estudio, con valores de 0.9969 F1-Score, 0.9950 mAP@0.5 y 0.9807 mAP@0.5:0.95. Este hallazgo contrasta con estudios previos, pero coincide con Parekh [84], quien informó que los modelos YOLOv8 superaron a RT-DETR en la detección de objetos de tráfico dentro de sistemas avanzados de asistencia al conductor. Estos altos valores obtenidos por el modelo muestran que el éxito de la clasificación se encuentra a un nivel utilizable en líneas de producción. El modelo también superó a otras versiones de YOLO. En términos de velocidad, si bien YOLOv8n es el modelo que realiza la inferencia más rápida, con un valor MITF de 6.1 ms/imagen, YOLOv8s se destaca como el modelo ideal en términos de velocidad, con un valor MITF de 10.3 ms/imagen. Ambos modelos desarrollados en el marco del estudio operaron con un tiempo medio de inferencia (MITF) bajo y mostraron un rendimiento que facilita su integración en sistemas de automatización industrial. Estos resultados respaldan la selección de YOLOv8s como el modelo más eficaz para la clasificación en tiempo real de piezas de pollo en procesos de envasado, gracias a su superior rendimiento tanto en precisión como en velocidad de inferencia.

Se ha demostrado que el modelo desarrollado en este estudio puede aplicarse de forma eficiente y eficaz en procesos de detección y clasificación

de objetos en tiempo real basados en aprendizaje profundo en la industria alimentaria. Se prevé que se obtendrán los siguientes beneficios utilizando este modelo en plantas de producción de pollo:

- Reducir la necesidad de intervención manual mediante la automatización del proceso de separación de piezas de pollo.
- Minimizar los errores causados por la clasificación manual.
- Reducir y eliminar el desperdicio de mano de obra y el desperdicio de empaques defectuosos.
- Aumentar el grado de automatización de los procesos de producción en plantas de pollo.
- Reducir las quejas de los clientes debido a empaques defectuosos y aumentar su satisfacción.
- Dirigir la fuerza laboral a áreas donde pueda utilizarse de manera más eficiente.
- Aumentar la velocidad y la eficiencia del proceso de producción y reducir los costos de producción.

El modelo desarrollado en este estudio ha demostrado un rendimiento utilizable en líneas industriales de envasado de pollo con alta precisión y un tiempo de inferencia reducido. Sin embargo, el modelo presenta algunas limitaciones. En primer lugar, el conjunto de datos se recopiló en un ambiente experimental creado de forma similar a una línea de producción, no en una línea de producción real. Los cambios en el equipo, el ángulo de la cámara, los sistemas de imagen y, especialmente, las condiciones de iluminación utilizadas en el ambiente de producción real pueden afectar el rendimiento del modelo. Por ejemplo, factores como la poca luz, las sombras o los reflejos pueden afectar negativamente la precisión de detección. Además, las dificultades en la implementación del modelo en situaciones reales incluyen factores como la detección de objetos que se mueven rápidamente en la línea de producción y la garantía de una integración adecuada del hardware. El trabajo futuro se centrará en el entrenamiento del modelo con datos recopilados en ambientes de producción reales, lo que aumentará su robustez ante condiciones de iluminación variables y optimizará su integración en procesos de producción reales. Otra limitación del estudio es que el fondo en la línea de producción examinada es regular, limpio y uniforme. Por lo tanto, el modelo desarrollado en este estudio se entrenó y evaluó mediante la creación de un ambiente experimental que simulara este ambiente de producción real. Se puede crear una nueva línea de investigación en futuros estudios para examinar el efecto de diferentes condiciones ambientales en el rendimiento del modelo. Además, este modelo, que ha demostrado su eficacia en la detección de partes de pollo antes del envasado, puede entrenarse en estudios futuros con conjuntos de datos recopilados de diferentes procesos de envasado, como pato o pavo, y su rendimiento puede investigarse para fines de clasificación automática.

Contribuciones del autor: Conceptualization, D.S., O.T. and M.S.; methodology, D.S., O.T. and M.S.; software, D.S., M.S., D.D.D. and A.K.; validation, D.S., O.T., M.S., D.D.D., R.Y. and A.K.; formal analysis, D.S., O.T., M.S. and R.Y.; investigation, D.S., O.T., M.S. and R.Y.; data curation, D.S., M.S. and D.D.D.; writing – original draft preparation, D.S., O.T., M.S., D.D.D.,

R.Y. and A.K.; writing—review and editing, D.S., O.T., M.S., D.D.D., R.Y. and A.K.; visualization, M.S., D.D.D. and A.K.; supervision, O.T. and R.Y.; project administration, O.T.; funding acquisition, D.S. and O.T. Todos los autores han leído y aceptado la versión publicada del manuscrito.

Fondos: Esta investigación fue financiada por el Proyecto Estudiantil 2209-A TUBITAK (Consejo de Investigación Científica y Tecnológica de Turquía), número de subvención 1919B012335040.

Declaración de disponibilidad de datos: Los datos presentados en este estudio están disponibles a solicitud del autor correspondiente.

Conflictos de intereses: Los autores declaran no tener conflictos de intereses.

Abreviaciones

En este manuscrito se utilizan las siguientes abreviaturas:

AIFI	Interacción de características intraescala basada en la atención
ANN	Redes neuronales artificiales
AP	Precisión media
AUC	Área bajo la curva característica de operación del receptor
C2F	Cuello de botella de CSP con 2 convoluciones
CCD	Dispositivo de carga acoplada
CCFM	Fusión de características entre escalas basada en CNN
CIOU	Intersección completa sobre la Unión
CNN	Red neuronal convolucional
CN _{TP}	Número de piezas de pollo detectadas correctamente por el modelo
CN _{TN}	Número de piezas de pollo que faltan correctamente
CN _{FP}	Número de piezas de pollo detectado incorrectamente
DETR	Transformador de detección
DFL	Pérdida centrada en la distribución
FPN	Red piramidal de características
FPS	Fotogramas por segundo
GFLOPs	Giga operaciones de punto flotante por segundo
GPU	Unidad de procesamiento de gráficos
IOU	Intersección sobre la Unión
mAP	Precisión media promedio
MITF	Tiempo medio de inferencia por fotograma
PAN	Red de agregación de rutas
PR	Precisión-recuperación
R-CNN	Redes neuronales convolucionales basadas en regiones
RT-DETR	Transformador de detección en tiempo real
SPPF	Agrupación piramidal espacial rápida
YOLOv4	Sólo miras una vez versión 4
YOLOv5	Sólo miras una vez versión 5
YOLOv5x	Sólo miras una vez versión 5-extra grande

Referencias

1. Omrak, H. Türkiye ocupa el décimo lugar en el mundo en producción de carne de ave. *Agric. For. J.* **2021**. Available online: <http://www.turktarim.gov.tr/Haber/702/turkiye-kanatli-eti-uretiminde-dunyada-10--sirada#:~:text=Sekt%C3%B6r,%2015%20bin%20adet%20kay%C4%B1tl%C4%B1,ise%20d%C3%BCnyada%207%E2%80%99nci%20s%C4%B1radad%C4%B1r> (Consultado el 22 de febrero de 2025). (In Turkish)
2. Türkiye Statistical Institute. Disponible en línea: <https://data.tuik.gov.tr/Bulten/Index?p=Kumes-Hayvanciligi-Uretimi-Ocak-2024-53562> (Consultado el 21 de noviembre de 2024). (In Turkish)

3. Statista. Producción mundial de carne de pollo 2012-2024. Available online: <https://www.statista.com/statistics/237637/production-of-poultry-meat-worldwide-since-1990/#:~:text=This%20statistic%20depicts%20chicken%20meat,million%20metric%20tons%20by%202024> (consultado el 21 de noviembre de 2024).
4. Cooreman-Algoed, M.; Boone, L.; Taelman, S.E.; Van Hemelryck, S.; Brunson, A.; Dewulf, J. Impacto del comportamiento del consumidor en el perfil de sostenibilidad ambiental de las cadenas de producción y consumo de alimentos: un estudio de caso sobre la carne de pollo. *Resour. Conserv. Recycl.* **2022**, *178*, 106089. [CrossRef]
5. Marcinkowska-Lesiak, M.; Zdanowska-Saśiadek, Z.; Stelmasiak, A.; Damaziak, K.; Michalczyk, M.; Poławska, E.; Wyrwicz, J.; Wierzbicka, A. Efecto del método de envasado y el tiempo de almacenamiento en frío en la calidad de la carne de pollo. *CyTA-J. Food* **2016**, *14*, 41–46. [CrossRef]
6. Teimouri, N.; Omid, M.; Mollazadeh, K.; Mousazadeh, H.; Alimardani, R.; Karstoft, H. Separación y clasificación en línea de porciones de pollo utilizando un enfoque de modelado inteligente basado en visión robusto. *Biosyst. Eng.* **2018**, *167*, 8–20. [CrossRef]
7. Nayyar, A.; Kumar, A. (Eds.) *Una hoja de ruta hacia la Industria 4.0: Producción inteligente, negocios inteligentes y desarrollo sostenible*; Springer Nature: Berlin/Heidelberg, Germany, 2019; ISBN 978-3-030-14543-9.
8. Chen, Y.W.; Shiu, J.M. Una implementación de los algoritmos de la familia YOLO para clasificar la calidad del producto para la metalización de acrilonitrilo butadieno estireno. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 8257–8269. [CrossRef] [PubMed]
9. Kulkarni, U.; Patil, A.; Devaranavadi, R.; Devagiri, S.B.; Pamali, S.K.; Ujawane, R. Control de calidad basado en visión de ejes tubulares mediante arquitectura DNN. En las actas de la Web de Conferencias ITM 2023., Gujarat, India, 28–29 April 2023; EDP Sciences. Volume 53, p. 02009. [CrossRef]
10. Chetoui, M.; Akhloufi, M.A. Inspección de calidad basada en modelos de detección de objetos mediante una CNN profunda. En las Actas de la Decimosexta Conferencia Internacional sobre Control de Calidad, por Artificial Vision, Albi, France, 6–8 June 2023; Volume 12749, pp. 65–72. [CrossRef]
11. Ardic, O.; Cetinel, G. Inspección de piezas de motor en tiempo real basada en aprendizaje profundo con aplicación robótica colaborativa. *IEEE Access* **2024**. [CrossRef]
12. Akgül, İ. Un nuevo método de aprendizaje profundo para detectar defectos en la superficie de la pantalla de un teléfono móvil basado en visión artificial. *Sak. Univ. J. Sci.* **2023**, *27*, 442–451. [CrossRef]
13. Alpdemir, M.N. Detección de defectos pseudosupervisada mediante autocodificadores convolucionales profundos y robustos. *Sak. Univ. J. Comput. Inf. Sci.* **2022**, *5*, 385–403. [CrossRef]
14. Güngör, M.A. Un nuevo método de detección de defectos superficiales basado en gradientes para baldosas cerámicas. *Sak. Univ. J. Sci.* **2022**, *26*, 1159–1169. [CrossRef]
15. Fan, J.; Zheng, P.; Li, S. Comprensión holística de la escena basada en la visión hacia la colaboración proactiva entre humanos y robots. *Robot. Comput.- Integr. Manuf.* **2022**, *75*, 102304. [CrossRef]
16. Zhang, R.; Lv, J.; Li, J.; Bao, J.; Zheng, P.; Peng, T. Un enfoque basado en gráficos y habilitado mediante aprendizaje de refuerzo para operaciones de ensamblaje colaborativo entre humanos y robots adaptativos. *J. Manuf. Syst.* **2022**, *63*, 491–503. [CrossRef]
17. Li, C.; Zheng, P.; Yin, Y.; Pang, Y.M.; Huo, S. Un enfoque basado en aprendizaje de refuerzo profundo asistido por RA para una interacción segura entre humanos y robots desde el punto de vista cognitivo mutuo. *Robot. Comput.-Integr. Manuf.* **2023**, *80*, 102471. [CrossRef]
18. Banafian, N.; Fesharakifard, R.; Menhaj, M.B. Seguimiento preciso de la costura en soldadura robótica mediante un enfoque de procesamiento de imágenes mejorado. *Int. J. Adv. Manuf. Technol.* **2021**, *114*, 251–270. [CrossRef]
19. Rout, A.; Deepak, B.B.V.L.; Biswal, B.B.; Mahanta, G.B. Detección de cordones de soldadura, búsqueda y configuración de parámetros de proceso para variar el espacio de soldadura mediante el uso de láser y sensor de visión en soldadura por arco robótico. *IEEE Trans. Ind. Electron.* **2021**, *69*, 622–632. [CrossRef]
20. Chen, C.; Chen, T.; Cai, Z.; Zeng, C.; Jin, X. Un modelo visual jerárquico para la guía automática de soldadura por arco mediante robots. *Ind. Robot. Int. J. Robot. Res. Appl.* **2023**, *50*, 299–313. [CrossRef]
21. Susto, G.A.; Schirru, A.; Pampuri, S.; McLoone, S.; Beghi, A. Aprendizaje automático para mantenimiento predictivo: un enfoque de clasificadores múltiples. *IEEE Trans. Ind. Inform.* **2014**, *11*, 812–820. [CrossRef]
22. Paolanti, M.; Romeo, L.; Felicetti, A.; Mancini, A.; Frontoni, E.; Loncarski, J. Enfoque de aprendizaje automático para el mantenimiento predictivo en la industria 4.0. En las actas de la 14.ª Conferencia internacional IEEE/ASME de 2018 sobre sistemas y aplicaciones mecatrónicas e integrados (MESA), Oulu, Finland, 2–4 July 2018. [CrossRef]
23. Liu, C.; Zhu, H.; Tang, D.; Nie, Q.; Zhou, T.; Wang, L.; Song, Y. Probandos un enfoque de mantenimiento predictivo inteligente con aprendizaje profundo y realidad aumentada para máquinas herramienta en la fabricación habilitada para IoT. *Robot. Comput.- Integr. Manuf.* **2022**, *77*, 102357. [CrossRef]

24. Zhuang, L.; Xu, A.; Wang, X.L. Un marco de mantenimiento predictivo impulsado por pronósticos basado en el aprendizaje profundo bayesiano. *Reliab. Eng. Syst. Saf.* **2023**, *234*, 109181. [\[CrossRef\]](#)
25. Knoll, D.; Prüglermeier, M.; Reinhart, G. Predicción de futuros procesos logísticos de entrada mediante aprendizaje automático. *Procedia CIRP* **2016**, *52*, 145–150. [\[CrossRef\]](#)
26. Yu, X.; Liao, X.; Li, W.; Liu, X.; Tao, Z. Control de automatización logística basado en algoritmos de aprendizaje automático. *Clust. Comput.* **2019**, *22*, 14003–14011. [\[CrossRef\]](#)
27. Abosuliman, S.S.; Almagrabi, A.O. Interacción humano-computadora asistida por visión artificial para la gestión logística mediante aprendizaje profundo. *Comput. Electr. Eng.* **2021**, *96*, 107555. [\[CrossRef\]](#)
28. Gregory, S.; Singh, U.; Gray, J.; Hobbs, J. Un flujo de trabajo de visión artificial para el seguimiento automático de inventarios a gran escala. En las Actas de la Conferencia ACM del Sudeste de 2021., New York, NY, USA, 15–17 April 2021; pp. 100–107. [\[CrossRef\]](#)
29. Denizhan, B.; Yıldırım, E.; Akkan, Ö. Un problema de preparación de pedidos en un centro médico utilizando un algoritmo genético. *Processes* **2024**, *13*, 22. [\[CrossRef\]](#)
30. Zhafran, F.; Ningrum, E.S.; Tamara, M.N.; Kusumawati, E. Sistema de visión artificial para la detección de equipos de protección individual (EPI) mediante redes neuronales convolucionales. Actas del Simposio Internacional de Electrónica (IES) de 2019., Surabaya, Indonesia, 27–28 September 2019. [\[CrossRef\]](#)
31. Khandelwal, P.; Khandelwal, A.; Agarwal, S.; Thomas, D.; Xavier, N.; Raghuraman, A. Uso de la visión artificial para mejorar la seguridad de la fuerza laboral en la fabricación en un mundo pos-COVID. *arXiv* **2020**, arXiv:2005.05287. [\[CrossRef\]](#)
32. Cheng, J.P.; Wong, P.K.Y.; Luo, H.; Wang, M.; Leung, P.H. Monitoreo basado en visión del cumplimiento de la seguridad del sitio basado en la reidentificación de los trabajadores y la clasificación del equipo de protección personal. *Autom. Constr.* **2022**, *139*, 104312. [\[CrossRef\]](#)
33. Zhang, H.; Ma, Z.; Li, X. Rs-detr: Un modelo mejorado de detección de objetos por teledetección basado en rt-detr. *Appl. Sci.* **2024**, *14*, 10331. [\[CrossRef\]](#)
34. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Hacia la detección de objetos en tiempo real con redes de propuesta de regiones. Hacia la detección de objetos en tiempo real con redes de propuesta de regiones. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1137–1149.
35. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. Solo se mira una vez: Detección unificada de objetos en tiempo real. En las Actas de la Conferencia IEEE sobre Visión por Computador y Reconocimiento de Patrones, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
36. Güney, E.; Bayilmis, C.; Çakan, B. Una implementación de detección de señales de tráfico y objetos viales en tiempo real basada en plataformas GPU móviles. *IEEE Access* **2022**, *10*, 86191–86203. [\[CrossRef\]](#)
37. Talaat, F.M.; ZainEldin, H. Un enfoque mejorado de detección de incendios basado en YOLO-v8 para ciudades inteligentes. *Neural Comput. Appl.* **2023**, *35*, 20939–20954. [\[CrossRef\]](#)
38. Inui, A.; Mifune, Y.; Nishimoto, H.; Mukohara, S.; Fukuda, S.; Kato, T.; Furukawa, T.; Tanaka, S.; Kusunose, M.; Takigami, S.; et al. Detección de TOC de codo en la imagen de ultrasonido mediante inteligencia artificial utilizando YOLOv8. *Appl. Sci.* **2023**, *13*, 7623. [\[CrossRef\]](#)
39. Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. Detección de objetos de extremo a extremo con transformadores. En la Conferencia Europea de Visión Artificial; Springer International Publishing: Cham, Switzerland, 2020; pp. 213–229. [\[CrossRef\]](#)
40. Nguyen, D.; Hoang, V.D.; Le, V.T.L. V-DETR: Transformador puro para la detección integral de objetos. En la Conferencia Asiática sobre Sistemas Inteligentes de Información y Bases de Datos; Springer Nature: Singapur, 2024; pp. 120–131. [\[CrossRef\]](#)
41. Zhao, X.; Song, Y. Detección de barcos mejorados con YOLOv8 mejorado con MobileViT y GSConv. *Electronics* **2023**, *12*, 4666. [\[CrossRef\]](#)
42. Wang, A.; Xu, Y.; Wang, H.; Wu, Z.; Wei, Z. CDE-DETR: Un método de detección de objetos mediante teledetección de alta resolución, de extremo a extremo y en tiempo real basado en RT-DETR. En las actas del Simposio Internacional de Geociencias y Teledetección del IEEE IGARSS 2024-2024, Athens, Greece, 7–12 July 2024.
43. Asmara, R.A.; Hasanah, Q.; Rahutomo, F.; Rohadi, E.; Siradjuddin, I.; Ronilaya, F.; Handayani, A.N. Identificación de la frescura de la carne de pollo mediante colores y texturas. En las Actas de la 7.ª Conferencia Internacional Conjunta sobre Informática, Electrónica y Visión (ICIEV) de 2018 y la 2.ª Conferencia Internacional sobre Imagenología, Visión y Reconocimiento de Patrones (icIVPR) de 2018, Kitakyushu, Japan, 25–29 June 2018. [\[CrossRef\]](#)
44. Putra, G.B.; Prakasa, E. Clasificación de la frescura de la carne de pollo mediante algoritmos de redes neuronales convolucionales. En las Actas de la Conferencia Internacional sobre Innovación e Inteligencia para la Informática, la Computación y las Tecnologías (3ICT) de 2020, Sakheer, Bahrain, 20–21 December 2020; pp. 1–6. [\[CrossRef\]](#)
45. Garcia, M.B.P.; Labuac, E.A.; Hortinela, C.C., IV. Clasificación de la frescura de la carne de pollo según la arquitectura vgg16. En

- las Actas de la Conferencia Internacional IEEE sobre Inteligencia Artificial en Ingeniería y Tecnología (IICAET) de 2022, Kota Kinabalu, Malaysia, 13–15 September 2022; pp. 1–6. [CrossRef]
46. Nyalala, I.; Okinda, C.; Makange, N.; Korohou, T.; Chao, Q.; Nyalala, L.; Zhang, J.; Zuo, Y.; Yousaf, K.; Liu, C.; et al. Estimación del peso en línea de canales y cortes de pollos de engorde mediante un sistema de visión artificial. *Poult. Sci.* **2021**, *100*, 101474. [CrossRef]
 47. You, M.; Liu, J.; Zhang, J.; Xv, M.; He, D. Un nuevo método de evaluación de la calidad de la carne de pollo basado en la localización de tarjetas de color y la corrección del color. *IEEE Access* **2020**, *8*, 170093–170100. [CrossRef]
 48. Salma, S.; Habib, M.; Tannouche, A.; Ounejjar, Y. Poultry Clasificación de carne utilizando el modelo preentrenado MobileNetV2. *Rev. D'intelligence Artif.* **2023**, *37*, 275–280. [CrossRef]
 49. Chen, Y.; Peng, X.; Cai, L.; Jiao, M.; Fu, D.; Xu, C.C.; Zhang, P. Investigación sobre clasificación y detección automática de partes de pollo basada en algoritmo de aprendizaje profundo. *J. Food Sci.* **2023**, *88*, 4180–4193. [CrossRef]
 50. Peng, X.; Xu, C.; Zhang, P.; Fu, D.; Chen, Y.; Hu, Z. Detección de clasificación por visión artificial de partes de pollo basada en un transformador Swin optimizado. *CyTA-J. Food* **2024**, *22*, 2347480. [CrossRef]
 51. Tamang, S.; Sen, B.; Pradhan, A.; Sharma, K.; Singh, V.K. Mejorando la seguridad frente a la COVID-19: Explorando la detección de objetos con yolov8 para una clasificación precisa de mascarillas faciales. *Int. J. Intell. Syst. Appl. Eng.* **2023**, *11*, 892–897. Available online: <https://ijisae.org/index.php/IJISAE/article/view/2966> (accessed on 20 February 2025).
 52. Bawankule, R.; Gaikwad, V.; Kulkarni, I.; Kulkarni, S.; Jadhav, A.; Ranjan, N. Detección visual de residuos mediante YOLOv8. En las Actas de la Conferencia Internacional sobre Computación Sostenible y Sistemas Inteligentes (ICSCSS) de 2023, Coimbatore, India, 14–16 June 2023; pp. 869–873. [CrossRef]
 53. Jun, E.L.T.; Tham, M.L.; Kwan, B.H. Análisis comparativo de RT-DETR y YOLOv8 para la detección de objetos aéreos en zonas urbanas. En las actas de la Conferencia Internacional IEEE sobre Control Automático y Sistemas Inteligentes (I2CACIS) de 2024. Shah Alam, Malaysia, 29 June 2024; pp. 340–345. [CrossRef]
 54. Guemas, E.; Routier, B.; Ghelfenstein-Ferreira, T.; Cordier, C.; Hartuis, S.; Marion, B.; Bertout, S.; Varlet-Marie, E.; Costa, D.; Pasquier, G. Reconocimiento automático a nivel de paciente de cuatro especies de Plasmodium en un frotis de sangre fino mediante un algoritmo de detección de objetos con transformador de detección en tiempo real (RT-DETR): una prueba de concepto y evaluación. *Microbiol. Spectr.* **2024**, *12*, e01440-23. [CrossRef] [PubMed]
 55. Jocher, G.; Chaurasia, A.; Qiu, J. YOLO by Ultralytics. 2023. Disponible en línea: <https://docs.ultralytics.com/models/yolov8> (accessed on 20 February 2025).
 56. Yu, Z.; Wan, L.; Yousaf, K.; Lin, H.; Zhang, J.; Jiao, H.; Yan, G.; Song, Z.; Tian, F. Un algoritmo de mejora para la detección de características de la cabeza de pollos enjaulados basada en una red neuronal de migración consistente cíclica. *Poult. Sci.* **2024**, *103*, 103663. [CrossRef] [PubMed]
 57. Dumitriu, A.; Tatui, F.; Miron, F.; Ionescu, R.T.; Timofte, R. Segmentación de corrientes de resaca: Un nuevo punto de referencia y resultados de referencia de yolov8. En las Actas de la Conferencia IEEE/CVF sobre Visión Artificial y Reconocimiento de Patrones, Vancouver, BC, Canada, 17–24 June 2023; pp. 1261–1271.
 58. Aishwarya, N.; Kumar, V. Clasificación de la madurez del plátano con CNN profunda en NVIDIA Jetson Xavier AGX. En las actas de la 7.ª Conferencia Internacional sobre I-SMAC (IoT en redes sociales, móviles, analítica y nube) de 2023 (I-SMAC), Kirtipur, Nepal, 11–13 October 2023; pp. 663–668. [CrossRef]
 59. Soylyu, E.; Soylyu, T. Una comparación del rendimiento de los modelos YOLOv8 para la detección de señales de tráfico en la competencia de vehículos autónomos a escala real Robotaxi. *Multimed. Tools Appl.* **2024**, *83*, 25005–25035. [CrossRef]
 60. Zhai, X.; Huang, Z.; Li, T.; Liu, H.; Wang, S. YOLO-Drone: Una red YOLOv8 optimizada para la detección de pequeños objetos UAV. *Electronics* **2023**, *12*, 3664. [CrossRef]
 61. Kumari, S.; Gautam, A.; Basak, S.; Saxena, N. Método de aprendizaje profundo basado en Yolov8 para la detección de baches. En las Actas de la Conferencia Internacional IEEE de 2023 sobre Visión Artificial e Inteligencia Artificial (CVMI), Gwalior, India, 10–11 December 2023; pp. 1–6. [CrossRef]
 62. Pereira, G.A. Detección de caídas para configuraciones industriales utilizando variantes de yolov8. *arXiv* **2024**, arXiv:2408.04605. [CrossRef]
 63. Zhao, Y.; Lv, W.; Xu, S.; Wei, J.; Wang, G.; Dang, Q.; Liu, Y.; Chen, J. Detsr superó a Yolov8 en la detección de objetos en tiempo real. En las Actas de la Conferencia IEEE/CVF sobre Visión Artificial y Reconocimiento de Patrones, Seattle, WA, USA, 16–22 June 2024; pp. 16965–16974.
 64. Lu, L.L.; Li, X.L.; Wu, Y.W.; Chen, B.C. RT-DETR mejorado para la detección de señales de tráfico: Precisión en objetos pequeños y diseño ligero. Research Square [Preimpresión]. 12 de noviembre de 2024. Disponible en línea: <https://www.researchsquare.com/article/rs-5351138/v1> (accessed on 20 February 2025).
 65. Liu, M.; Wang, H.; Du, L.; Ji, F.; Zhang, M. Bearing-detr: Un modelo de aprendizaje profundo liviano para la detección de defectos en rodamientos basados en rt-detr. *Sensors* **2024**, *24*, 4262. [CrossRef] [PubMed]
 66. Dai, L.; Wang, D.; Song, F.; Yang, H. Método de detección de grietas en puentes de hormigón basado en un modelo RT-DETR

- mejorado. En las actas de la 3.^a Conferencia Internacional sobre Robótica, Inteligencia Artificial y Control Inteligente (RAIIC) de 2024, Mianyang, China, 5–7 July 2024; pp. 172–175.
67. Liu, Y.; Cao, Y.; Sun, Y. Investigación sobre un método de reconocimiento de defectos ferroviarios basado en el modelo RT-DETR mejorado. En las actas de la 5.^a Conferencia Internacional sobre Ingeniería Informática y Aplicaciones (ICCEA) de 2024, Hangzhou, China, 12–14 April 2024; pp. 1464–1468. [\[CrossRef\]](#)
 68. Li, X.; Cai, M.; Tan, X.; Yin, C.; Chen, W.; Liu, Z.; Wen, J.; Han, Y. Una red de transformadores eficiente para detectar pollos de múltiples escalas en ambientes complejos de cría en libertad a través de RT-DETR mejorado. *Comput. Electron. Agric.* **2024**, *224*, 109160. [\[CrossRef\]](#)
 69. Cao, X.; Wang, H.; Wang, X.; Hu, B. DFS-DETR: Detector sensible a características detalladas para la detección de objetos pequeños en imágenes aéreas mediante transformador. *Electronics* **2024**, *13*, 3404. [\[CrossRef\]](#)
 70. Tang, S.; Yan, W. Utilización del modelo RT-DETR para la estimación de calorías de frutas a partir de imágenes digitales. *Información* **2024**, *15*, 469. [\[CrossRef\]](#)
 71. Yu, C.; Chen, X. Detección de defectos de ahuellamiento ferroviario basada en RT-DETR mejorado. *J. Real-Time Image Process.* **2024**, *21*, 146. [\[CrossRef\]](#)
 72. Bayraktar, E.; Basarkan, M.E.; Celebi, N. Un marco de UAV de bajo costo para la detección y el recuento de plantas ornamentales en la naturaleza. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 1–11. [\[CrossRef\]](#)
 73. Mamdouh, N.; Khattab, A. Marco de aprendizaje profundo basado en YOLO para la detección y el recuento de la mosca del olivo. *IEEE Access* **2021**, *9*, 84252–84262. [\[CrossRef\]](#)
 74. Ghorbanzadeh, O.; Crivellari, A.; Ghamisi, P.; Shahabi, H.; Blaschke, T. Una evaluación integral de la transferibilidad de U-Net y ResU-Net para la detección de deslizamientos de tierra a partir de datos de Sentinel-2 (áreas de estudio de caso de Taiwán, China y Japón). *Sci. Rep.* **2021**, *11*, 14629. [\[CrossRef\]](#)
 75. Torkul, O.; Selvi, I.H.; Sisci, M.; Diren, D.D. Un nuevo modelo para el reconocimiento de tareas de ensamblaje: un estudio de caso del sistema de producción Seru. *IEEE Access* **2024**. [\[CrossRef\]](#)
 76. Jang, W.S.; Kim, S.; Yun, P.S.; Jang, H.S.; Seong, Y.W.; Yang, H.S.; Chang, J.S. Detección precisa de implantes dentales y tejido periimplantario mediante aprendizaje por transferencia de R-CNN más rápida: un estudio de precisión diagnóstica. *BMC Oral Health* **2022**, *22*, 591. [\[CrossRef\]](#)
 77. Qadir, H.A.; Shin, Y.; Solhusvik, J.; Bergsland, J.; Aabakken, L.; Balasingham, I. Hacia la detección de pólipos en tiempo real utilizando CNN completas para la predicción de formas gaussianas 2D. *Med. Image Anal.* **2021**, *68*, 101897. [\[CrossRef\]](#) [\[PubMed\]](#)
 78. Su, Y.; Cheng, B.; Cai, Y. Detección y reconocimiento de fragmentos de medicina tradicional china basados en YOLOv8. En las actas de la 6.^a Conferencia Internacional sobre Tecnologías de la Información y la Comunicación Electrónica (ICEICT) del IEEE de 2023, Qingdao, China, 21–24 July 2023; pp. 214–217. [\[CrossRef\]](#)
 79. Kingma, D.P.; Ba, J. Adam: Un método para la optimización estocástica. *arXiv* **2014**, arXiv:1412.6980. [\[CrossRef\]](#)
 80. Reis, D.; Kupec, J.; Hong, J.; Daoudi, A. Detección de objetos voladores en tiempo real con YOLOv8. *arXiv* **2023**, arXiv:2305.09972. [\[CrossRef\]](#)
 81. Ding, J.; Zhang, J.; Zhan, Z.; Tang, X.; Wang, X. Un método preciso y eficiente para la detección de edificios colapsados en imágenes UAV posteriores a terremotos basado en el algoritmo NMS mejorado y el R-CNN más rápido. *Remote Sens.* **2022**, *14*, 663. [\[CrossRef\]](#)
 82. Jin, M.; Zhang, J. Investigación sobre detección de logotipos de vehículos a microescala basada en un transformador de detección en tiempo real (RT-DETR). *Sensors* **2024**, *24*, 6987. [\[CrossRef\]](#)
 83. Wu, M.; Qiu, Y.; Wang, W.; Su, X.; Cao, Y.; Bai, Y. RT-DETR mejorado y su aplicación a la detección de la madurez de la fruta. *Front. Plant Sci.* **2025**, *16*, 1423682. [\[CrossRef\]](#)
 84. Parekh, A. Análisis comparativo de YOLOv8 y RT-DETR para la detección de objetos en tiempo real en sistemas avanzados de asistencia a la conducción. Trabajo de fin de máster. The University of Western Ontario, London, ON, Canada, 2025.

Descargo de responsabilidad/Nota del editor: Las declaraciones, opiniones y datos contenidos en todas las publicaciones son responsabilidad exclusiva de los autores y colaboradores, y no de MDPI ni de sus editores. MDPI y sus editores no se responsabilizan de ningún daño a personas o bienes que resulte de las ideas, métodos, instrucciones o productos mencionados en el contenido.