

# Implementación de Redes Neuronales YOLO para la Detección de la Abertura de Cajas en Sistemas de Paquetería

**Ryan A. LEÓN LEÓN**

Universidad Privada del Norte  
Trujillo, Perú

**Jeisin I. AGUILAR ZAVALA**

Universidad Privada del Norte  
Trujillo, Perú

**Diana M. CHÁVEZ VÁSQUEZ**

Universidad Privada del Norte  
Trujillo, Perú

**Erika L. BRIONES FLORES**

Universidad Privada del Norte  
Trujillo, Perú

**Glenda H. MEJIA RUBIO**

Universidad Privada del Norte  
Trujillo, Perú

**Jhon A. HUARINGA ROLDAN**

Universidad Privada del Norte  
Trujillo, Perú

## *Implementation of YOLO Neural Networks for the Detection of Box Opening in Packaging Systems*

**ABSTRACT:** *The aim of this study is the implementation of YOLOv8 neural networks to enhance the automatic detection of unauthorized openings in boxes within parcel delivery systems. The methodology included the creation of a varied dataset from proprietary images and a public database, managed and labeled in Roboflow, ensuring adequate representation of real-world lighting conditions and viewing angles. YOLOv8, an advanced version of the algorithm, was used in Google Colab for training, utilizing image preprocessing and postprocessing techniques that improved the model's accuracy and reliability in detecting openings. The results showed a 98% accuracy in detecting openings. The application of YOLOv8 for detecting unauthorized openings in boxes provides a modern and precise solution to enhance security and quality control in the supply chain.*

**Keywords:** *opening detection, YOLOv8, parcel delivery systems, YOLOv8, Roboflow, Google Colab.*

**RESUMEN:** El presente estudio tiene como objetivo la implementación de redes neuronales YOLOv8 para mejorar la detección automática de aperturas no autorizadas en cajas dentro de sistemas de paquetería. La metodología incluyó la creación de un conjunto de datos variado a partir de imágenes propias y una base de datos pública, gestionada y etiquetada en Roboflow, asegurando una representación adecuada de condiciones reales de iluminación y ángulos de visión. Se empleó YOLOv8, una versión avanzada del algoritmo, en Google Colab para el entrenamiento, utilizando técnicas de preprocesamiento y postprocesamiento de imágenes que mejoraron la precisión y confiabilidad del modelo en la detección de aperturas. Los resultados mostraron una precisión del 98% en la detección de aperturas. la aplicación de YOLOv8 para la detección de

aperturas no autorizadas en cajas ofrece una solución moderna y precisa para mejorar la seguridad y control de calidad en la cadena de suministro.

**Palabras clave:** detección de aperturas, YOLOv8, sistemas de paquetería, YOLOv8, Roboflow, Google Colab.

## I. INTRODUCCIÓN

La manipulación indebida en los sistemas de paquetería es una causa frecuente de pérdidas económicas, agravada por el crecimiento del comercio electrónico. Estudios, como los de IATA y BSI, destacan que el robo de carga y la apertura no autorizada de paquetes representa miles de millones de dólares en pérdidas anuales, afectando tanto a empresas de logística como a comerciantes. Actualmente, muchas empresas dependen de inspecciones visuales esporádicas y registros manuales, que limitan la detección precisa y en tiempo real de manipulaciones no autorizadas, permitiendo que los robos o daños pasen desapercibidos hasta la entrega. La implementación de redes neuronales YOLO (You Only Look Once) ofrece una solución avanzada a este problema, permitiendo una detección automatizada y en tiempo real de aperturas de cajas, mejorando significativamente la capacidad para identificar manipulaciones no autorizadas y asegurar la integridad de los paquetes en tránsito. Las redes neuronales convolucionales (CNN) son un ejemplo de tipos exitosos de modelos de aprendizaje profundo [1]. Las CNN se componen de muchas capas. Cada capa transforma su entrada aplicando pequeños filtros de convolución cuyos pesos se aprenden de manera supervisada utilizando grandes muestras de entrenamiento [2]. A pesar de su dependencia de grandes conjuntos de entrenamiento, el aprendizaje profundo ha comenzado recientemente a ofrecer resultados interesantes y a mostrar impacto en muchas aplicaciones [3]. Esto se debe principalmente a la disponibilidad

de grandes conjuntos de datos de entrenamiento y hardware potente que puede acelerar el proceso de entrenamiento [4]. YOLO (solo miras una vez) ha emergido como un modelo esencial para la detección de objetos, destacándose especialmente en aplicaciones en tiempo real debido a su notable precisión y rapidez [5,6]. La detección de objetos ha avanzado considerablemente en el campo de la visión por computadora, ofreciendo métodos confiables para localizar e identificar objetos en imágenes y videos [7]. YOLO está diseñado para proporcionar las coordenadas de los objetos dentro de una imagen en la que ha sido entrenado, junto con una puntuación de confianza que refleja la fiabilidad de la predicción. La familia YOLO ha evolucionado a través de múltiples iteraciones desde su inicio, y cada nueva versión posterior se basa en las anteriores para abordar los desafíos de detección y mejorar el rendimiento general [8,9]. La incorporación de CNN y modelos como YOLO en la visión por computadora ha logrado un progreso significativo en la automatización de la detección y clasificación de objetos [10]. Estos avances se deben a la mayor disponibilidad de datos y al incremento en la potencia computacional, que permiten entrenar modelos más sofisticados y exactos. La evolución de YOLO y otros modelos de detección subraya la tendencia hacia una mayor eficiencia y precisión en aplicaciones en tiempo real, reafirmando la importancia de las CNN como herramientas esenciales en la visión artificial contemporánea [11,12]. En este contexto, se desarrolló un sistema en tiempo real para la detección de defectos en el empaquetado utilizando el algoritmo YOLO, logrando una precisión del 91% y mejorando tanto la eficiencia como el control de calidad del empaquetado [13]. En otro estudio, se empleó un modelo Bilinear-VGG16 mejorado para la inspección de la calidad de paquetes de cigarrillos. Este modelo, entrenado con imágenes de paquetes defectuosos y calificados, alcanzó una precisión del 96.3% en la detección de defectos, optimizando el proceso de inspección de calidad [14]. Adicionalmente, se presentó un proceso de monitoreo en línea para el empaquetado de alimentos que utiliza un modelo de red neuronal profunda. Al integrar técnicas avanzadas de convolución profunda y ajustes en VGG16, junto con redes neuronales convolucionales, aprendizaje por transferencia y redes neuronales adaptativas, se logró una precisión de reconocimiento del 97% [15]. Sin embargo, es importante destacar que hay pocos estudios que aborden la implementación de redes neuronales YOLO específicamente para la detección de la apertura de cajas en sistemas de paquetería, lo que subraya una oportunidad significativa para la investigación y desarrollo en esta área. Lo mencionado anteriormente destaca la importancia de una detección precisa y eficiente de defectos en el empaquetado es crucial para garantizar la calidad y la integridad de los productos en sistemas de paquetería. La apertura incorrecta o deficiente de cajas puede llevar a problemas significativos en la cadena de suministro, afectando la seguridad del producto y la satisfacción del cliente. Sin embargo, la mayoría de los métodos tradicionales de inspección de empaques son manuales y están sujetos a errores humanos, lo que limita su eficacia y eficiencia. El objetivo de esta contribución es la implementación de redes neuronales. En total, se seleccionaron 1,645 imágenes, que fueron sometidas a un riguroso proceso de curación. Imágenes de baja calidad, con problemas de enfoque o iluminación, fueron eliminadas, Asegurando así que el conjunto de datos final estuviera compuesto únicamente por imágenes claras y representativas [17]. Una vez depurado, el dataset fue dividido de manera estratégica: el 92% de las imágenes se destinó al entrenamiento del modelo, el 6% fue utilizado para validación, y el 2% se reservó para pruebas. Esta distribución permitió que el modelo tuviera una cantidad suficiente de datos para aprender las

YOLO para mejorar la detección automática de la apertura de cajas en sistemas de paquetería. Además. Los objetivos específicos de esta contribución incluyen alcanzar una precisión mínima del 92% en la detección automática de aperturas no autorizadas en sistemas de paquetería mediante redes neuronales YOLO, asegurando que los errores de detección sean mínimos. Adicionalmente, se trabajará en la adaptación y optimización del modelo YOLO para garantizar su robustez en condiciones variadas de iluminación y ángulos de visión, logrando una fiabilidad alta en entornos de producción reales, incluso en escenarios logísticos complejos.

## II. METODOLOGÍA

El proyecto de detección de apertura de cajas en paquetería se basó en un conjunto de datos con imágenes propias y de una base pública, gestionadas en Roboflow. Se utilizó una cámara Canon EOS M50 (24.1 MP) para capturar detalles esenciales, bajo luz natural y con trípode para estabilidad. Las imágenes se tomaron a 50 cm - 1 m de distancia, con ángulos frontales y laterales. La combinación de datasets permitió mayor diversidad en tipos de cajas, ángulos e iluminación, mejorando la adaptabilidad del modelo a distintas condiciones de manipulación en sistemas de paquetería.



Fig. 1 Proceso de etiquetado – Roboflow

En Roboflow, se llevó a cabo un meticuloso proceso de etiquetado manual (Fig. 1) por cada imagen que va a ser parte del dataset, ya sea para validación, prueba o entrenamiento, en el que cada imagen fue clasificada de acuerdo con si la caja estaba abierta o cerrada. Esta plataforma fue fundamental no solo para facilitar el etiquetado, sino también para permitir la gestión eficiente de las imágenes y asegurar que los datos estuvieran correctamente organizados y preparados para su uso en el modelo de detección [16].

Características visuales asociadas a las cajas abiertas y cerradas, mientras que el conjunto de validación y test aseguraban una correcta evaluación de su desempeño en escenarios no vistos previamente. Además de la partición del dataset, se realizaron ajustes visuales mediante técnicas de aumento de datos, como rotaciones y ajustes de brillo, para enriquecer aún más el conjunto de datos y mejorar la robustez del modelo. Este enfoque no solo aumentó la variabilidad de las imágenes, sino que también mejoró la capacidad del modelo para generalizar en diferentes condiciones de operación. En este estudio se empleó YOLOv8,

una de las versiones más avanzadas en detección de objetos, destacada por su alta precisión y velocidad de procesamiento. Este modelo utiliza una arquitectura basada en redes neuronales convolucionales profundas, permitiendo la detección de múltiples defectos en una sola pasada a través de la red, lo que lo hace más eficiente en comparación con otros enfoques que requieren varios pasos de procesamiento.

La elección de YOLOv8 está respaldada por su rendimiento superior en tareas de visión artificial, lo que ha sido documentado en investigaciones recientes que subrayan su ventaja sobre versiones anteriores y otros modelos de detección [18-20]. El procedimiento comienza con la introducción de imágenes de cajas en el sistema. Durante la fase inicial, conocida como Backbone, se utilizan varias capas convolucionales para extraer características esenciales como bordes, texturas y formas relacionadas con los defectos en las cajas (Fig. 2). Estas capas analizan la imagen aplicando filtros que recorren la matriz de píxeles, generando mapas de características que permiten identificar elementos importantes, como abolladuras, rasguños o signos de deterioro.

En esta fase, se implementan capas C2f (Cross Stage Partial Networks), que optimizan la transferencia de información entre capas, mejorando la extracción de características visuales clave al reducir redundancias y aumentar la eficiencia en la representación de características. Además, se emplea el bloque SPPF (Spatial Pyramid Pooling Fast), El modelo implementa un mecanismo avanzado de pooling en múltiples escalas, permitiendo reducir la dimensionalidad de los datos y optimizar la eficiencia computacional sin comprometer la calidad de las detecciones. Esta estrategia facilita la extracción de características clave a diferentes niveles de la imagen, lo que mejora la capacidad del modelo para identificar patrones en distintas resoluciones. Además, al reducir la cantidad de datos procesados por las capas posteriores, se minimiza el costo computacional, permitiendo una ejecución más rápida y eficiente, especialmente en dispositivos con recursos limitados. Como resultado, el modelo mantiene un equilibrio óptimo entre precisión y velocidad, lo que lo hace ideal para aplicaciones en tiempo real y sistemas embebidos.

En la etapa siguiente, denominada Neck, se combinan las características extraídas mediante técnicas de upsampling (aumento de la resolución de mapas de características) y concatenación de información de diferentes niveles de resolución. Este proceso permite integrar tanto detalles finos como patrones globales, asegurando que el modelo pueda identificar defectos pequeños y localizados (como una abolladura) así como características más amplias (como un deterioro extenso en la caja).

Además, en esta etapa se utilizan bloques CSP (Cross Stage Partial) para mejorar la capacidad de generalización del modelo, asegurando que las características combinadas sean representativas de las posibles variaciones en las imágenes. Este enfoque ha demostrado ser altamente efectivo en investigaciones previas, especialmente en aplicaciones industriales donde es crucial detectar características complejas y multiescalares. Finalmente, en la fase de Head o predicción, YOLOv8 procesa la información para producir resultados precisos. Esto incluye la generación de cuadros delimitadores (bounding boxes) que indican las posiciones exactas de los defectos, la clasificación del tipo de defecto identificado (como abolladuras, rasguños o

deterioro) y el cálculo del nivel de confianza asociado a cada detección.



Fig.2 Operaciones convolucionales

La salida generada por el sistema es visual, mostrando las áreas afectadas con cuadros delimitadores y etiquetas que indican tanto el tipo de defecto como su nivel de confianza. Este método no solo asegura la identificación rápida y precisa de defectos, sino que también permite la interpretación visual de los resultados, facilitando la validación de las detecciones. Este método es crucial para garantizar la correcta identificación de defectos antes de que las cajas avancen a la siguiente fase del proceso productivo. La rapidez y precisión de YOLOv8 lo convierten en una herramienta fundamental para implementar sistemas de clasificación automatizada altamente eficientes en entornos industriales [21].

En este estudio, se utilizó Google Colab como plataforma para entrenar el modelo YOLOv8m, aprovechando su infraestructura de alto rendimiento y sus recursos especializados en redes neuronales, en particular las GPUs. Se empleó una GPU T4 con 15 GB de memoria y soporte para CUDA. Gracias a la capacidad de esta GPU, se logró una rápida convergencia del modelo, optimizando así los tiempos de entrenamiento. Después de configurar el entorno en Google Colab, se integraron las bibliotecas necesarias, como torch y la implementación de YOLOv8. PyTorch, reconocida por su versatilidad y facilidad de uso, es una herramienta muy utilizada en la investigación de redes neuronales profundas, lo que la convierte en una opción ideal para este proyecto.

Para el entrenamiento del modelo en Google Colab, se cargaron datos previamente etiquetados de cajas con defectos y se organizaron en el entorno de trabajo. Se ajustaron parámetros clave del modelo YOLOv8, como el tamaño del lote, la tasa de aprendizaje y el número de épocas, fijándose en 450 épocas para garantizar un entrenamiento exhaustivo. Durante esta fase, se monitorizaron constantemente métricas como la pérdida y la precisión en el conjunto de validación, lo que permitió ajustar los hiperparámetros y mejorar el rendimiento del modelo a medida que avanzaba el entrenamiento (Fig. 3).

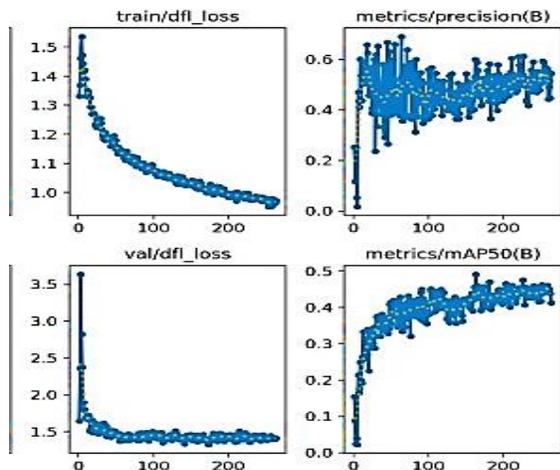


Fig. 3 Redimiendo del en el tiempo – se toma como referencia la precisión y la función perdida a través de las épocas

El uso de estas herramientas y configuraciones permitió desarrollar un modelo optimizado para la detección de defectos en cajas, asegurando tanto la precisión en la identificación de abolladuras, rasguños y deterioros, como la eficiencia en el tiempo de procesamiento, factores cruciales para su implementación en entornos productivos. La función de pérdida es el elemento central que guía el entrenamiento de un modelo como YOLOv8. En este proyecto, su objetivo es minimizar los errores cometidos a los tipos de cajas, optimizando así la precisión del modelo. Esta función cuantifica las discrepancias entre las predicciones del modelo y las etiquetas reales, lo que permite ajustar los pesos de la red neuronal para mejorar el rendimiento.

$$L = L_{\text{coord}} + L_{\text{conf}} + L_{\text{cls}} \quad (1)$$

La función de pérdida en YOLOv8 está compuesta por tres términos principales, cada uno con una contribución específica al aprendizaje del modelo.  $L_{\text{coord}}$  mejora la precisión espacial al garantizar que las cajas delimitadoras predichas se ajusten correctamente al objeto.  $L_{\text{conf}}$  refuerza la capacidad del modelo para diferenciar entre regiones con y sin objetos, reduciendo así falsos positivos y negativos. Finalmente,  $L_{\text{cls}}$  optimiza la clasificación de los objetos, asegurando que las etiquetas asignadas a cada detección sean precisas.

Estos tres componentes trabajan en conjunto para maximizar la efectividad del modelo en tareas complejas de detección y clasificación. La tasa de aprendizaje controla la magnitud de los ajustes realizados a los pesos del modelo en cada iteración. En YOLOv8, generalmente se utiliza una tasa de aprendizaje dinámica ajustada mediante un programador de ciclos (OneCycle Scheduler). La ecuación para la actualización de los pesos con la tasa de aprendizaje

$$w_{t+1} = w_t - \eta \cdot \frac{\partial L}{\partial w_t} \quad (2)$$

En la ecuación, los términos representan lo siguiente,  $w_t$  es el peso en la iteración actual,  $\eta$  es la tasa de aprendizaje,  $\frac{\partial L}{\partial w_t}$  es el gradiente de la pérdida respecto al peso. Esta ecuación describe el proceso mediante el cual los pesos del modelo se ajustan iterativamente durante el entrenamiento, tomando en cuenta la

tasa de aprendizaje y el gradiente de la función de pérdida. Este mecanismo garantiza que el modelo refine sus predicciones progresivamente, minimizando el error y mejorando su capacidad de generalización. Al concluir el entrenamiento, se evaluó el modelo con un conjunto de datos de prueba independiente, permitiendo analizar su desempeño en escenarios no vistos previamente. Se emplearon métricas de validación, como la precisión, recall y F1-score, para verificar que el modelo no presentaba sobreajuste y que mantenía un equilibrio adecuado entre sensibilidad y especificidad.

Este procedimiento fue esencial para validar la efectividad del modelo en la detección de defectos en cajas, tales como abolladuras, rasguños y deterioro (Fig. 4). Además, se realizó una inspección visual de los resultados mediante una matriz de confusión, asegurando que las predicciones fueran consistentes con las etiquetas reales.

Para mejorar aún más el rendimiento, se consideró la posibilidad de realizar ajustes en los hiperparámetros, como la optimización del learning rate y la exploración de diferentes arquitecturas de red. De esta manera, se busca garantizar que el sistema mantenga una alta precisión incluso en condiciones de iluminación o ángulos de captura adversos, favoreciendo su implementación en entornos industriales de inspección automatizada.



Fig. 4 Evaluación del modelo

Para llevar a cabo la implementación del sistema de detección de defectos en cajas, se utilizó Python dentro del entorno de desarrollo Visual Studio. Después de configurar el entorno con las dependencias requeridas, como opencv-python para el manejo de imágenes y pytorch para ejecutar el modelo YOLOv8, se buscó optimizar la precisión del modelo a través de técnicas adicionales de visión por computadora [22].

El modelo YOLOv8, que ya había sido entrenado para identificar defectos como abolladuras, rasguños y deterioro en las cajas, se integró en un script de Python destinado al procesamiento de imágenes. Para aumentar la precisión y minimizar los falsos positivos, se aplicaron varias técnicas de preprocesamiento y postprocesamiento de imágenes. En primer lugar, se implementaron diversas técnicas de preprocesamiento de imágenes, como la normalización del histograma, con el objetivo de mejorar la visibilidad en imágenes que presentaban condiciones de iluminación desiguales. Este ajuste permitió al modelo YOLOv8 detectar de manera más precisa las aberturas de las cajas, resaltando las zonas clave que podrían haber pasado

desapercibidas en imágenes con bajo contraste. De toda la lectura podemos representar el proceso del algoritmo de detección de cajas con YOLOv8, iniciando con la carga del modelo y la activación de la cámara. Luego, se captura un frame y, si es válido, se procede a detectar y rastrear objetos. Si la confianza de detección supera el 94%, se clasifica como "Caja", de lo contrario, como "sin Caja". En caso de detección positiva, los resultados se dibujan en el frame para visualización. Finalmente, el sistema verifica si la tecla "Q" ha sido presionada; si no, el proceso se repite, de lo contrario, finaliza la ejecución. Para incrementar la precisión en la detección de las aberturas, se aplicó un filtro Gaussiano (Ecuación 3) que permitió reducir el ruido en las imágenes sin comprometer detalles importantes. En esta ecuación, el valor de entrada es la ubicación del pixel ( $x, y$ ) en la imagen, mientras que  $\sigma$  representa la desviación estándar, que determina la intensidad del suavizado aplicado al entorno del pixel. Como resultado, el filtro genera una nueva imagen donde el ruido aleatorio se atenúa, mejorando la claridad en las regiones de interés.

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

Además, durante la fase de postprocesamiento, se utilizaron técnicas avanzadas para afinar las predicciones generadas por YOLOv8. Después de que el modelo generaba los cuadros delimitadores y las etiquetas correspondientes, se empleó el análisis de contornos y la transformada de Hough (Ecuación 4) para perfeccionar los bordes de las aberturas detectadas. En esta ecuación, los valores de entrada incluyen las coordenadas ( $x, y$ ) de los puntos en la imagen, mientras que los parámetros de salida,  $\rho$  y  $\theta$ , describen la representación en coordenadas polares de las líneas detectadas. Este proceso permite identificar y ajustar los bordes lineales de las aberturas, logrando una delimitación más precisa de las áreas relevantes.

$$\rho = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (4)$$

Con el fin de minimizar falsos positivos y negativos, se incorporó un filtro basado en las dimensiones de los cuadros delimitadores. Los valores de entrada a este filtro son las dimensiones del cuadro generado por el modelo, y el resultado del filtro elimina aquellas detecciones cuyas dimensiones no coincidan con los tamaños esperados de las aberturas de las cajas. Este enfoque fue fundamental para aumentar la confiabilidad del modelo en la detección de aberturas pequeñas o parcialmente visibles. Finalmente, se utilizaron técnicas de dilatación y erosión para suavizar los bordes de las detecciones. Estas técnicas reciben como entrada la imagen binaria resultante de las detecciones previas y generan una imagen refinada donde los bordes son más uniformes. Este procesamiento adicional ayudó a reducir la imprecisión en zonas donde las aberturas podían confundirse con el fondo o situarse en los límites de las cajas. Estas mejoras en el procesamiento de imágenes y en los ajustes del modelo contribuyeron significativamente a aumentar la precisión general del sistema, optimizando la detección de las aberturas en las cajas y reduciendo al mínimo la incidencia de falsos positivos y errores en la identificación.

### III. RESULTADOS

Se logró una precisión del 98% en la detección de aberturas de cajas. Este nivel de precisión destaca la eficacia del modelo YOLOv8 al identificar manipulaciones o aberturas en empaques de forma precisa y en tiempo real. Los resultados obtenidos superan los de estudios previos en inspección de empaquetado, que alcanzaban precisiones inferiores, demostrando que las

optimizaciones específicas realizadas en la configuración de YOLO han sido claves para optimizar el desempeño del sistema y mejorar significativamente la exactitud en la detección de aberturas en sistemas de paquetería.

### IV. DISCUSIÓN

Nuestro estudio, que alcanzó una precisión del 98% en la detección de aberturas de cajas, supera los resultados de investigaciones previas en detección de defectos en empaquetado. En un estudio anterior, se utilizó YOLO para detección de defectos en tiempo real, logrando una precisión del 91%, lo que evidenció mejoras en la eficiencia y el control de calidad del empaquetado, aunque con una precisión significativamente inferior a la de nuestro modelo [13]. Esto sugiere que los ajustes y adaptaciones de YOLO en nuestro estudio han optimizado su desempeño para la detección específica de aberturas, logrando una mayor exactitud.

Otro estudio utilizó un modelo Bilinear-VGG16 mejorado para la inspección de calidad de paquetes de cigarrillos, alcanzando una precisión del 96.3% [14]. Aunque esta cifra es competitiva, sigue por debajo del 98% obtenido en nuestra investigación, lo que indica que nuestro enfoque resulta aún más confiable para aplicaciones de control de integridad de paquetes en tiempo real. Por su parte, un sistema de monitoreo en línea para el empaquetado de alimentos, que empleó una red neuronal profunda con técnicas avanzadas de VGG16, aprendizaje por transferencia y redes adaptativas, logró una precisión del 97% [15]. Aunque cercano a nuestro resultado, nuestro sistema de detección supera esta cifra, destacando la robustez y precisión de nuestro enfoque, posiblemente debido a las optimizaciones específicas para detectar aberturas y manipulaciones en cajas.

La precisión obtenida en nuestro estudio también pone de manifiesto el poder de las redes convolucionales profundas (CNN) y del modelo YOLOv8 en tareas de visión artificial específicas. Las CNN han demostrado una gran capacidad para extraer características complejas a partir de datos de imágenes, lo que es esencial para identificar patrones sutiles en objetos o empaques. Su estructura de capas convolucionales permite capturar detalles relevantes en distintas escalas y orientaciones, lo que facilita la detección precisa de anomalías. En combinación con la arquitectura YOLO, que procesa las imágenes en tiempo real, las CNN logran resultados sobresalientes en cuanto a velocidad y precisión [23]. YOLOv8, en particular, incorpora mejoras significativas respecto a versiones anteriores, como un diseño más eficiente y optimizaciones en la selección de anclas y en la red neuronal base. Estas mejoras permiten un procesamiento más rápido y efectivo en escenarios de control de calidad, donde cada fracción de segunda cuenta. Además, YOLOv8 puede adaptarse mejor a diversas resoluciones y condiciones de iluminación, lo que amplía su aplicabilidad en entornos industriales [24, 25]. En nuestro caso, las optimizaciones implementadas permitieron que el modelo se centrara en la detección específica de aberturas de cajas, con una precisión que establece un nuevo estándar en la inspección automatizada de empaques.

### V. CONCLUSIÓN

En conclusión, la implementación de redes neuronales YOLO para la detección de aberturas no autorizadas en cajas demuestra ser una solución efectiva y moderna para abordar los desafíos de seguridad y control de calidad en la cadena de suministro. A

diferencia de los métodos de inspección manual, que son lentos y propensos a errores, el sistema automatizado basado en YOLO ofrece precisión, eficiencia y confiabilidad, alcanzando el objetivo de una precisión mínima del 98 % en la identificación de aperturas indebidas. Además, al adaptar el modelo para operar bajo condiciones diversas de iluminación y ángulos de visión, se asegura un desempeño robusto en entornos reales de producción y logística. Este avance no solo reduce el riesgo de manipulaciones y daños en los productos, sino que también optimiza la operatividad en la cadena de suministro, contribuyendo a una mayor satisfacción del cliente y a una gestión logística más segura y eficiente.

Los resultados obtenidos en este estudio demuestran que la implementación de redes neuronales YOLOv8 para la detección de aperturas no autorizadas en cajas dentro de sistemas de paquetería representa un avance significativo en el ámbito de la seguridad y el control de calidad en la cadena de suministro. En comparación con los métodos tradicionales basados en inspecciones manuales, que suelen ser ineficientes y propensos a errores humanos, el enfoque automatizado basado en visión por computadora y aprendizaje profundo ofrece una alternativa más precisa, rápida y escalable.

El modelo propuesto alcanzó una precisión del 98%, superando a estudios previos que han empleado arquitecturas de redes neuronales como VGG16 y Faster R-CNN en la inspección de empaques, cuyos niveles de precisión oscilaron entre el 91% y 97%. Esta mejora en la exactitud no solo reduce la posibilidad de errores en la identificación de aperturas indebidas, sino que también fortalece los protocolos de seguridad en la cadena de suministro al permitir la detección en tiempo real de posibles manipulaciones. La detección temprana de aperturas no autorizadas evita pérdidas económicas y mejora la trazabilidad de los paquetes durante el proceso logístico, lo que resulta fundamental en sectores como el comercio electrónico, la logística industrial y la distribución farmacéutica. El éxito del sistema propuesto radica en la combinación de un conjunto de datos amplio y variado, técnicas avanzadas de preprocesamiento y postprocesamiento de imágenes, y la implementación de ajustes en la arquitectura de YOLOv8. La integración de técnicas como normalización del histograma, filtrado gaussiano y transformada de Hough permitió optimizar la detección de contornos y reducir falsos positivos y negativos, mejorando así la fiabilidad del modelo en entornos operativos reales. Además, la capacidad del modelo para adaptarse a diferentes condiciones de iluminación y ángulos de visión refuerza su aplicabilidad en entornos dinámicos, donde las condiciones de captura de imágenes pueden variar considerablemente.

El uso de modelos de visión por computadora basados en aprendizaje profundo abre nuevas oportunidades para la automatización de procesos en la industria de la paquetería y logística. La capacidad de detectar aperturas de cajas con alta precisión no solo beneficia a empresas de transporte y distribución, sino que también puede ser extrapolada a otros sectores que requieran inspección de empaques, como la industria manufacturera, el sector alimenticio y la farmacéutica. Además, la integración de estos sistemas con dispositivos IoT y tecnologías de blockchain podría proporcionar soluciones avanzadas de trazabilidad y auditoría en tiempo real, garantizando la seguridad de los productos desde su origen hasta su destino. A pesar del alto desempeño del modelo, existen desafíos que pueden ser abordados en investigaciones futuras. En primer lugar, la implementación en tiempo real a gran escala

requerirá optimizaciones en términos de eficiencia computacional, especialmente en sistemas con recursos limitados. Asimismo, el uso de técnicas de aprendizaje incremental y federado podría mejorar la adaptabilidad del modelo a nuevos entornos sin necesidad de reentrenamientos extensivos. Finalmente, el desarrollo de modelos híbridos que combinen redes neuronales convolucionales con modelos basados en transformar para visión artificial podría ofrecer mejoras adicionales en la precisión y robustez del sistema. En definitiva, la investigación confirma que el uso de redes neuronales YOLOv8 para la detección de aperturas en cajas no solo es viable, sino que también establece un nuevo estándar en la automatización del control de calidad en la industria logística. La precisión alcanzada, combinada con la capacidad de procesamiento en tiempo real, convierte a esta solución en una herramienta indispensable para optimizar la seguridad y la eficiencia en la gestión de paquetes. A medida que la tecnología avanza, la integración de técnicas de inteligencia artificial en los sistemas logísticos seguirá desempeñando un papel clave en la transformación digital de la industria.

## VI. REFERENCIAS

- [1] Pinaya, Walter Hugo Lopez, et al. Redes neuronales convolucionales. En Machine learning . Academic Press, 2020. p. 173-191. <https://doi.org/10.1016/B978-0-12-815739-8.00010-9>
- [2] Habib, Gousia; QURESHI, Shaima. Optimization and acceleration of convolutional neural networks: A survey. Journal of King Saud University-Computer and Information Sciences, 2022, vol. 34, no 7, p. 4244-4268. <https://doi.org/10.1016/j.jksuci.2020.10.004>
- [3] Veličković, Petar. Everything is connected: Graph neural networks. Current Opinion in Structural Biology, 2023, vol. 79, p. 102538. <https://doi.org/10.1016/j.sbi.2023.102538>
- [4] Valueva, Maria V., et al. Aplicación del sistema de números de residuos para reducir los costos de hardware de la implementación de la red neuronal convolucional. Matemáticas y computadoras en simulación , 2020, vol. 177, pág. 232-243. <https://doi.org/10.1016/j.matcom.2020.04.031>
- [5] Jiang, Peiyuan, et al. A Review of Yolo algorithm developments. Procedia computer science, 2022, vol. 199, p. 1066-1073. <https://doi.org/10.1016/j.procs.2022.01.135>
- [6] Diwan, Tausif; Anirudh, G.; Tembhurne, Jitendra V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. multimedia Tools and Applications, 2023, vol. 82, no 6, p. 9243-9275. <https://doi.org/10.1007/s11042-022-13644-y>
- [7] Chen, Wei, et al. A review of object detection: Datasets, performance evaluation, architecture, applications and current trends. Multimedia Tools and Applications, 2024, p. 1-59. <https://doi.org/10.1007/s11042-023-17949-4>
- [8] Vijayakumar, Ajantha, et al. Real-time visual intelligence for defect detection in pharmaceutical packaging. Scientific Reports, 2024, vol. 14, no 1, p. 18811. <https://doi.org/10.1038/s41598-024-69701-z>

- [9] Soviany, Petru; Ionescu, Radu Tudor. Optimizing the trade-off between single-stage and two-stage object detectors using image difficulty prediction. arXiv preprint arXiv:1803.08707, 2018. <https://doi.org/10.48550/arXiv.1803.08707>
- [10] He, Xing, et al. TPMN: Texture Prior-Aware Multi-Level Feature Fusion Network for Corrugated Cardboard Parcels Defect Detection. International Journal of Advanced Computer Science & Applications, 2024, vol. 15, no 2. doi: 10.14569/ijacsa.2024.0150284. Available in:
- [11] Essid, Oumayma; Laga, Hamid; Samir, Chafik. Automatic detection and classification of manufacturing defects in metal boxes using deep neural networks. PloS one, 2018, vol. 13, no 11, p. e0203192. <https://doi.org/10.1371/journal.pone.0203192>
- [12] H. Ding, Y. Zhang, X. Fu and Z. Li, "Based on Faster-RCNN for mobile phone packaging box defect detection," 2023 3rd International Symposium on Computer Technology and Information Science (ISCTIS), Chengdu, China, 2023, pp. 677-680, doi: 10.1109/ISCTIS58954.2023.10213193. Available in: <https://ieeexplore.ieee.org/abstract/document/10213193>
- [13] Pham DL, Chang TW. A YOLO-based real-time packaging defect detection system. Procedia Computer Science. 2023 Jan 1;217:886-94. <https://doi.org/10.1016/j.procs.2022.12.285>
- [14] Xu, Z., Guo, S., Li, Y., Wang, J., Ma, Y., Henna, L. Cigarette Packaging Quality Inspection Based on Convolutional Neural Network. 2022. Lecture Notes in Computer Science, vol 13338. Springer, Cham. [https://doi.org/10.1007/978-3-031-06794-5\\_49](https://doi.org/10.1007/978-3-031-06794-5_49)
- [15] Liu, Xiaobing. Computer-Aided Design of Hand-Drawn art Food Packaging Design Based on Deep Neural Network Model. Applied Mathematics and Nonlinear Sciences, 2023, vol. 8, no 2, p. 2043-2052. <https://doi.org/10.2478/amns.2023.1.00308>
- [16] Soni, Vama, et al. Introducing AOD 4: A dataset for air borne object detection. Data in Brief, 2024, vol. 56, p. 110801. <https://doi.org/10.1016/j.dib.2024.110801>
- [17] Chen, Wei, et al. Wall Cracks Detection in Aerial Images Using Improved Mask R-CNN. CMC-COMPUTERS MATERIALS & CONTINUA, 2022, vol. 73, no 1, p. 767-782. a Tech Science Press. doi: 10.32604/cmc.2022.028571. Available in: [https://cdn.techscience.cn/ueditor/files/cmc/TSP\\_CMC-73-1/TSP\\_CMC\\_28571/TSP\\_CMC\\_28571.pdf](https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC-73-1/TSP_CMC_28571/TSP_CMC_28571.pdf)
- [18] Hussain, Muhammad. Yolov1 to v8: Unveiling each variant—a comprehensive review of yolo. IEEE Access, 2024, vol. 12, p. 42816-42833. doi: 10.1109/ACCESS.2024.3378568. Available in : <https://ieeexplore.ieee.org/abstract/document/10473783>
- [19] Safaldin, Mukaram; Zaghdien, Nizar; Mejdoub, Mahmoud. An Improved YOLOv8 to Detect Moving Objects. IEEE Access, 2024. doi: 10.1109/ACCESS.2024.3393835. Available in: <https://ieeexplore.ieee.org/abstract/document/10508365>
- [20] Terven, Juan; Cordova, Diana-Margarita; Romero, Julio-Alejandro. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction, 2023, vol. 5, no 4, p. 1680-171. <https://doi.org/10.3390/make5040083>
- [21] M. Safaldin, N. Zaghdien y M. Mejdoub, "An Improved YOLOv8 to Detect Moving Objects", en Acceso IEEE12, pp. 59782-59806, 2024, doi: 10.1109/ACCESS.2024.3393835. <https://doi.org/10.1016/j.procs.2021.05.025>
- [22] Kumar, Ashwani; Srivastava, Sonam. Object detection system based on convolution neural networks using single shot multi-box detector. Procedia Computer Science, 2020, vol. 171, p. 2610-2617. <https://doi.org/10.1016/j.procs.2020.04.283>
- [23] Gu, Jiuxiang, et al. Recent advances in convolutional neural networks. Pattern recognition, 2018, vol. 77, p. 354-377. <https://doi.org/10.1016/j.patcog.2017.10.013>
- [24] Sohan, Mupparaju, et al. A review on yolov8 and its advancements. En International Conference on Data Intelligence and Cognitive Informatics. Springer, Singapore, 2024. p. 529-545. <https://doi.org/10.1016/j.patcog.2017.10.013>
- [25] Terven, Juan; Córdova-Esparza, Diana-Margarita; Romero-González, Julio-Alejandro. A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. Machine Learning and Knowledge Extraction, 2023, vol. 5, no 4, p. 1680-1716. <https://doi.org/10.3390/make5040083>