



Tipo de artículo: Artículos originales
Temática: Procesamiento de imágenes
Recibido: 18/05/2024 | Aceptado: 26/07/2024 | Publicado: 30/09/2024

Identificadores persistentes:
DOI: [10.48168/innosoft.s16.a160](https://doi.org/10.48168/innosoft.s16.a160)
ARK: [ark:/42411/s16/a160](https://nbn-resolving.org/ark:/42411/s16/a160)
PURL: [42411/s16/a160](https://nbn-resolving.org/urn:nbn:pe:ulasalle:innosoft-160)

Modelo Basado en YOLOv8 para la Detección de Residuos Sólidos

YOLOv8-Based Model for Solid Waste Detection

Rodrigo Alonso Guevara Saldaña^{1*}, Marcos Iván Díaz Tomás², Marcelino Torres Villanueva³

¹Universidad Nacional de Trujillo. t1053300621@unitru.edu.pe

²Universidad Nacional de Trujillo. t1533300221@unitru.edu.pe

³Universidad Nacional de Trujillo. mtorres@unitru.edu.pe

*Autor para correspondencia: t1533300621@unitru.edu.pe

Resumen

El principal motivo de este artículo fue la práctica de reconocimiento de objetos, utilizando la tecnología de Ultralytics, YOLOv8, aplicando el aprendizaje supervisado y otros métodos de machine learning. Se tomó en cuenta las definiciones de la detección de objetos y entrenamiento de modelos para clasificar los residuos sólidos para que estos sean después reciclados, para luego identificar manualmente cada clase de objeto con el etiquetador LabelImg, tomando en cuenta las posiciones de cada objeto en las imágenes. Se analizaron 1517 imágenes dándonos unos resultados excelentes y considerables.

Palabras claves: Detección, Aprendizaje profundo, Residuos Sólidos, YOLO

Abstract

The primary focus of this article was to employ Ultralytics technology, specifically YOLOv8, in object recognition. This involved utilizing supervised learning and other machine learning techniques. The article took into consideration the definitions of object detection and model training to effectively categorize solid waste, thereby facilitating recycling efforts. Following this, each object class was manually identified using the LabelImg tagger, considering the positions of the objects within the images. This approach led to the analysis of 1517 images and produced notably high-quality and significant results.

Keywords: *Detection, Deep Learning, Solid Waste, YOLO.*

Introducción

Muchas cosas que vemos a menudo podemos identificarlas sin problemas, es posible porque desde que tenemos razón nos enseñan a nombrar las cosas por un nombre conocido por nuestra comunidad que nos rodea. Tomando esta premisa como punto inicial, el crecimiento de las inteligencias artificiales y el desarrollo del aprendizaje

profundo; hablaremos sobre el tan conocido tema de la detección de objetos. La aplicación de estas tecnologías ha mostrado grandes resultados, por ejemplo, en sistemas de detección de peatones [1] se usa bastante esta definición, haciendo que el modelo de aprendizaje sea entrenado rigurosamente.

La detección de objetos utiliza el aprendizaje profundo y la visión por computadora o artificial para lograr identificar un objeto dentro de una imagen utilizando diferentes métodos o enfoques, la detección en tiempo real requiere de una precisión adecuada y tiempos de respuesta cortos, necesidades que métodos como YOLO, SSD o RetinaNet cuentan y, por lo tanto, los hacen adecuados para esta tarea [2]; ya que la detección de objetos puede llevarse a diversos rubros de las industrias, la detección de tumores [3] o la detección de objetos pequeños para imágenes aéreas [4].

Al momento de hablar sobre residuos sólidos nos referimos a los materiales (sólidos o semisólidos) inservibles que no tienen un uso directo y se tiene la necesidad de eliminarlos [5]. Es por ello que la gestión de estos se ha convertido en un tema de gran interés que se encuentra relacionado con muchos temas que guardan mucha relación con la problemática ambiental que nos está afectando a nosotros [6], es por ello que tenemos la necesidad de poder separar los tipos de residuos, ya sean papeles, cartón u objetos de plástico.

EL objetivo de este trabajo fue aportar soluciones tecnológicas a la gestión de residuos sólidos, y como dice [7], YOLOv8 es una red de detección de objetos en tiempo real, el cual se orienta a la regresión, para que así una sola red neuronal nos predijo cuadros delimitadores y probabilidades de clases asociadas.

Materiales y métodos o Metodología computacional

Hardware y Software

La computadora donde se realizó el trabajo cuenta con las siguientes especificaciones: Procesador Intel Core i7 6700, memoria RAM DDR4 de 8GB y una tarjeta gráfica NVIDIA RTX 2060 con 6GB de VRAM. El trabajo se realizó dentro del sistema operativo Windows 10, utilizando el lenguaje de programación Python en su versión 3.11.0, la librería Ultralytics YOLOv8 y sus dependencias junto con Pytorch con soporte para GPU.

Conjunto de Datos

Los elementos del conjunto de datos son clases, imágenes y anotaciones o etiquetas. Las clases son todos los diferentes objetos que el modelo deberá ser capaz de detectar. En las imágenes se encuentran las clases, los objetos deben tener diferentes características como color o tamaño para una mayor variedad en los datos. Las etiquetas son archivos de texto donde se encuentra la ubicación y tamaño de cada clase presente en una imagen.

Las clases son residuos sólidos comunes que pueden encontrarse en la mayoría de ambientes. Los residuos incluidos son de diferentes materiales y se encuentran en diferentes formas o presentaciones. A continuación, se presentan las clases utilizadas y una pequeña descripción:

- carton: Cajas de cartón.
- electrico_cable: Cables de corriente, USB, etc.
- electrico_chip: Circuitos PCB.
- electrico_pila: Pilas y baterías.
- madera: Tablones de madera.
- medicinal_guante: Guantes médicos desechables.
- medicinal_mascarilla: Mascarillas desechables.
- metal_contenedor: Envases de metal.
- metal_lata: Latas de bebidas de metal.
- metal_pieza: Piezas de metal como tornillos, alambre, cables, varas.
- organico_hoja: Hojas de árboles o plantas.
- papel_hoja: Hojas de papel intactas o arrugadas.
- papel_periodico: Periódicos.
- papel_vaso: Vasos de papel.
- plastico_bolsa: Bolsas de plástico.
- plastico_botella: Botellas de plástico.
- plastico_cubierto: Tenedores, cucharas o cuchillos de plástico.
- plastico_envase: Tapers, contenedores y demás envases de plástico.
- lastico_vaso: Vasos de plástico.
- spray_envase: Latas de spray o aerosoles.

- vidrio_botella: Botellas de bebidas de vidrio.

Las imágenes que conforman el conjunto de datos, algunas son provenientes del conjunto de datos TrashBox [8]. Las imágenes de las clases madera y organico_hoja se obtuvieron con el software WFdownloader [9], esta herramienta permite descargar grandes cantidades de imágenes de diversas páginas web automáticamente. El etiquetado de las imágenes recolectadas se realizó con el software LabelImg [10]. En cada imagen, se dibujó un cuadro delimitador o bounding box donde se encuentra una instancia de una clase, la herramienta se encarga de crear las etiquetas en formato YOLO [11], basándose en los cuadros delimitadores dibujados. En la Figura 1, se muestra una imagen donde se dibujaron los cuadros delimitadores para tres instancias de la clase metal_lata.

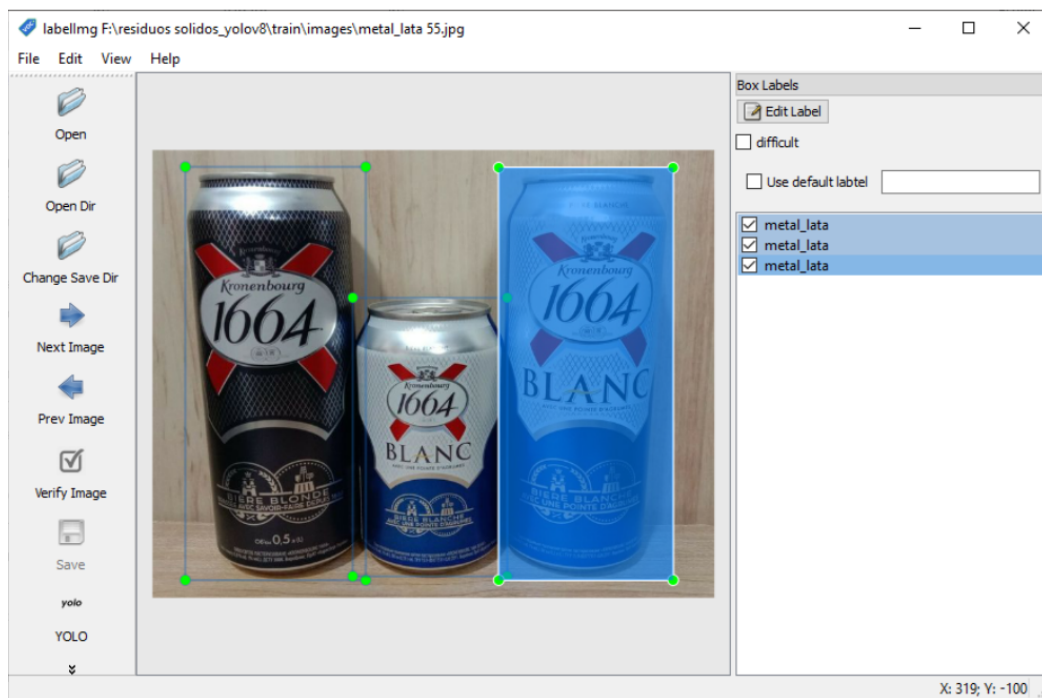


Figura 1. Herramienta LabelImg para etiquetar imágenes.

El conjunto de datos está conformado por 1517 imágenes con un total de 4860 instancias. Este conjunto de datos se dividió en dos partes, una destinada para entrenar el modelo y otra para validar el modelo. El conjunto de datos para entrenamiento constó de 1098 imágenes con 3514 instancias, y para validar, 419 imágenes con 1346 instancias.

YOLOv8 utiliza un archivo de configuración en formato .yaml para indicar las carpetas con los datos para

entrenar y validar [2], así como también las clases a utilizar, siguiendo la siguiente estructura:

- train: Carpeta con las imágenes y etiquetas para el entrenamiento del modelo.
- val: Carpeta con las imágenes y etiquetas para la validación del modelo.
- nc: Número de clases del conjunto de datos.
- names: Nombres de las clases utilizadas.

Métricas

En [2] - [12] - [13], se utilizan valores como IoU, AP y mAP para evaluar modelos basados en YOLO y otros modelos similares para la detección de objetos.

Según [14], IoU (Intersection over Union, o Intersección sobre Unión) es un valor que determina la superposición del cuadro delimitador detectado sobre el cuadro real. El valor P (Precision, o precisión) mide la cantidad de verdaderos positivos sobre todas las predicciones positivas. R (Recall, o rellamada) se encarga de calcular la proporción de verdaderos positivos sobre todos los positivos reales. AP (Average precision, o precisión promedio) es el área sobre la curva P/R (Precision and recall) de cada clase, es utilizado para medir la relación entre precisión y rellamada del modelo. mAP (Mean average precisión, o precisión promedio media) es similar a AP, pero se calcula para todas las clases dentro del conjunto de datos, sirve para evaluar el rendimiento general del modelo.

YOLOv8 utiliza los siguientes valores para la validación de modelos [15]:

- P: Capacidad para detectar los objetos correctamente.
- R: Capacidad de detectar todas las instancias de objetos en las imágenes.
- mAP50: Precisión promedio media para un valor IoU de 0,50.
- mAP50-95: Precisión promedio media calculada para valores IoU en el rango de 0,50 y 0,95.

Entrenamiento del Modelo

YOLOv8 tiene diferentes modelos pre entrenados para facilitar el entrenamiento de modelos con un conjunto de datos personalizado. Estos modelos fueron entrenados con el conjunto de datos COCO [16], las especificaciones se muestran en la Tabla 1.

Tabla 1. Comparación de Modelos YOLOv8 pre entrenados.

Modelo	Tamaño de Imagen (píxeles)	mAP ₅₀₋₉₅	Velocidad A100 (ms)	Parámetros (M)	FLOPS (B)
YOLOv8n	640	37,3	0,99	3,2	8,7
YOLOv8s	640	44,9	1,20	11,2	28,6
YOLOv8m	640	50,2	1,83	25,9	78,9
YOLOv8l	640	52,9	2,39	42,7	165,2
YOLOv8x	640	53,9	3,53	68,2	257,8

La principal diferencia entre estos modelos es el valor de mAP y la velocidad. Basándonos en estas cifras, modelos como YOLOv8l o YOLOv8x logran mayor precisión a costa de velocidad y requiriendo mayor potencia de cómputo.

Considerando las limitaciones de hardware y el tamaño del conjunto de datos, el modelo pre entrenado seleccionado para este trabajo fue YOLOv8s.

El entrenamiento de modelos basados en YOLOv8 puede hacerse a través de scripts en el lenguaje de programación Python, como también con una CLI (Interfaz de línea de comandos) [17]. Para este trabajo se utilizó la CLI, donde indicamos el archivo .yaml con las carpetas para los datos de entrenamiento, el modelo pre entrenado seleccionado y los hiper parámetros batch y epoch.

Los hiper parámetros son valores para controlar el aprendizaje de nuestro modelo. En [18], batch se define como el número de muestras procesadas antes de actualizar el modelo, y epoch es el número de pasadas sobre el conjunto de datos. Ambos parámetros son usados para actualizar el modelo, mejorando la precisión de este.

Validación

En [19] se explica que YOLOv8 tiene la capacidad de validar el modelo para analizar el rendimiento de este y poder mejorarlo. De la misma manera que el entrenamiento, se utilizó la herramienta CLI para la validación, donde indicamos el modelo ya entrenado con el conjunto de datos de residuos sólidos, el archivo .yaml donde se encuentra la carpeta con los datos de validación. Los resultados de la validación determinaran la eficiencia del modelo.

Resultados y discusión

Los resultados de la validación del modelo se muestran en la Tabla 2

Los resultados muestran que el modelo en general tiene una gran precisión para clases como `electrico_cable`, elec-

Tabla 2. Resultados de Validación del Modelo

Clases	Instancias	P	R	mAP50	mAP50-95
Todas	1346	0,97	0,99	0,991	0,909
carton	22	0,981	1	0,995	0,995
electrico_cable	22	0,97	1	0,995	0,94
electrico_chip	27	0,998	1	0,995	0,935
electrico_pila	62	0,98	1	0,995	0,915
madera	95	0,969	0,999	0,994	0,873
medicinal_guante	109	0,855	0,844	0,923	0,724
medicinal_mascarilla	30	0,948	1	0,995	0,927
metal_contenedor	39	0,963	1	0,995	0,972
metal_lata	135	0,964	1	0,994	0,919
metal_pieza	75	0,95	1	0,994	0,893
organico_hoja	159	0,992	0,952	0,979	0,781
papel_hoja	82	0,997	1	0,995	0,866
papel_periodico	45	0,937	0,988	0,993	0,917
papel_vaso	67	0,982	1	0,995	0,919
plastico_bolsa	23	0,981	1	0,995	0,984
plastico_botella	51	0,994	1	0,995	0,953
plastico_cubierto	88	0,988	1	0,995	0,848
plastico_envase	75	0,994	1	0,995	0,928
plastico_vaso	39	0,989	1	0,995	0,94
spray_envase	60	0,996	1	0,995	0,918
vidrio_botella	41	0,99	1	0,995	0,951

trico_chip, electrico_pila, plastico_bolsa, plastico_botella, plastico_cubierto, plastico_envase, plastico_vaso, spray_envase y vidrio_botella. La clase medicinal_guante es la única clase con precisión menor a 0,90, para mejorar este valor se pueden añadir más datos de esta clase en un futuro entrenamiento y así mejorar su variedad. En la Figura 2 se puede apreciar una instancia de la clase medicinal_guante detectada por el modelo.



Figura 2. Detección de medicinal_guante.

Conclusiones

Con una precisión del 0.90, el modelo entrenado puede facilitar la clasificación de residuos sólidos al implementarse en plantas de procesamiento de residuos sólido, evitando que trabajadores puedan verse afectados por entrar en contacto con residuos sólidos peligrosos para la salud, además de la automatización y mejorar la eficiencia. Aunque debido al tamaño del conjunto de datos, será necesario añadir más clases y así poder detectar todos los tipos de residuos sólidos que se puedan generar dentro de una ciudad.

Contribución de Autoría

Rodrigo Alonso Guevara Saldaña: [Conceptualización](#), [Investigación](#), [Metodología](#), [Software](#), [Validación](#), [Redacción - borrador original](#), [Administración de proyectos](#).

Marcos Iván Díaz Tomás: [Visualización](#), [Supervisión](#), [Curación de datos](#), [Escritura](#), [revisión y edición](#).

Marcelino Torres Villanueva: [Supervisión](#), [Escritura y revisión](#).

Referencias

- [1] M. Razzok, A. Badri, I. E. Mourabit, Y. Ruichek, and A. Sahel, "Pedestrian detection under weather conditions using conditional generative adversarial network," *Int J Artif Intell*, vol. 12, no. 4, pp. 1557–1568, 2023.

- [2] U. Sirisha, S. P. Praveen, P. N. Srinivasu, P. Barsocchi, and A. K. Bhoi, “Statistical analysis of design aspects of various yolo-based deep learning models for object detection,” *International Journal of Computational Intelligence Systems*, vol. 16, no. 1, p. 126, 2023.
- [3] M. G. Dinesh, N. Bacanin, S. S. Askar, and M. Abouhawwash, “Diagnostic ability of deep learning in detection of pancreatic tumour,” *Scientific Reports*, vol. 13, no. 1, p. 9725, 2023.
- [4] K. Niu and Y. Yan, “A small-object-detection model based on improved yolov8 for uav aerial images,” in *2023 2nd International Conference on Artificial Intelligence and Intelligent Information Processing (AIIIP)*. IEEE, 2023, pp. 57–60.
- [5] “Región natural año 5. residuos sólidos,” 2012. [Online]. Available: https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitaless/Est/Lib1756/cap05.pdf
- [6] A. F. M. Rendón, “Caracterización de residuos sólidos,” *Cuaderno activa*, vol. 4, pp. 67–72, 2012.
- [7] D. Horcajada Jiménez, “Metodología para la detección de objetos en imágenes basada en la librería yolo con aplicación a la detección de carros,” Master’s thesis, Unknown, 2021.
- [8] N. V. Kumsetty, A. B. Nekkare, S. Kamath S, and A. Kumar M, “Trashbox: Trash detection and classification using quantum transfer learning,” 2022.
- [9] “Wfdownloader app - free bulk image downloader and multi-purpose bulk downloader,” 2023. [Online]. Available: <https://www.wfdownloader.xyz/>
- [10] G. Boesch, “Labelimg for image annotation - viso.ai,” 2022. [Online]. Available: <https://viso.ai/computer-vision/labeling-for-image-annotation/>
- [11] Ultralytics, “Object detection datasets overview,” 2023. [Online]. Available: <https://docs.ultralytics.com/es/datasets/detect/#supported-datasets>
- [12] Y. M. Tang, W. T. Kuo, and C. K. Lee, “Real-time mixed reality (mr) and artificial intelligence (ai) object recognition integration for digital twin in industry 4.0,” *Scopus*, 2023.
- [13] J. Hui, “map (mean average precision) for object detection,” 2018. [Online]. Available: <https://jonathan-hui.medium.com/map-mean-average-precision-for-object-detection-45c121a31173>
- [14] Ultralytics, “Yolo performance metrics,” 2023. [Online]. Available: <https://docs.ultralytics.com/es/guides/yolo-performance-metrics/>
- [15] —, “Yolov8,” 2023. [Online]. Available: <https://docs.ultralytics.com/es/models/yolov8/#performance-metrics>

- [16] —, “Train,” 2023. [Online]. Available: <https://docs.ultralytics.com/es/modes/train/>
- [17] J. Brownlee, “What is the difference between a batch and an epoch in a neural network,” *Machine Learning Mastery*, vol. 20, 2018.
- [18] Ultralytics, “Val,” 2023. [Online]. Available: <https://docs.ultralytics.com/es/modes/val/>