

DETECCIÓN Y CLASIFICACIÓN DE DESPERDICIOS HUMANOS A TRAVÉS DE YOLOV8 Y UNA CÁMARA WEB

GARBAGE CLASSIFICATION DETECTION USING YOLOV8 AND A WEBCAM

Luis Edgar Alanís Carranza

Tecnológico Nacional de México / TES de Coacalco, México
luis.alanis@tesco.edu.mx

Adrián Israel Altamirano Arroyo

Tecnológico Nacional de México / TES de Coacalco, México
202121020@tesco.edu.mx

Erick Ricardo Jiménez Hernández

Tecnológico Nacional de México / TES de Coacalco, México
202120578@tesco.edu.mx

Iván Daniel Plasencia Gonzáles

Tecnológico Nacional de México / TES de Coacalco, México
i.plasencia@tesco.edu.mx

Jesús Axkana Fuentes Luna

Tecnológico Nacional de México / TES de Coacalco, México
202120236@tesco.edu.mx

Israel Alejandro Pérez Luna

Tecnológico Nacional de México / TES de Coacalco, México
202120118@tesco.edu.mx

Emiliano Sánchez Hernández

Tecnológico Nacional de México / TES de Coacalco, México
202120224@tesco.edu.mx

Recepción: 25/septiembre/2023

Aceptación: 24/abril/2024

Resumen

En la actualidad, la clasificación de basura no se lleva a cabo de manera adecuada, lo que ocasiona la pérdida de materia prima indispensable para la industria. Una gran cantidad de materiales reciclables no son reutilizados y ha provocado que cada año aumente la cantidad de residuos en diferentes partes del mundo.

Este estudio propone la implementación de YOLOv8 con el fin de detectar y clasificar los desperdicios humanos mediante una cámara web. El modelo fue entrenado mediante la utilización de una base de datos de diversos materiales en contenedores de basura.

Los resultados del entrenamiento revelaron una precisión general del 86% con una velocidad de inferencia de 50.2 *ms/imagen*. Este modelo puede detectar y clasificar correctamente la mayoría de los desechos humanos e implementarse en una cámara web con una velocidad de procesamiento por fotograma inferior a 51 *ms*.

Palabras Clave: Desperdicios humanos, Redes neuronales convolucionales, YOLOv8.

Abstract

Currently, the classification of garbage is not carried out adequately, which causes the loss of essential raw materials for the industry. A significant quantity of recyclable materials is not repurposed, resulting in an increase in waste volume every year in diverse regions of the globe.

This paper proposes the implementation of YOLOv8 to detect and classify garbage through a webcam. The model was trained by utilizing a database comprising diverse materials in garbage containers.

*The training results showed an overall accuracy of 86% with an inference speed of 50.2 *ms/image*. This model can correctly detect and classify most garbage and can be implemented on a webcam with a frame rate of less than 51 *ms*.*

Keywords: Convolutional Neural Network, Garbage, YOLOv8.

1. Introducción

En la actualidad, una considerable cantidad de residuos en el mundo carecen de la atención adecuada para su descomposición. Como resultado, los ecosistemas son afectados directamente por la enorme cantidad de basura acumulada.

La Asociación Internacional de Ingeniería Sanitaria y Ambiental (AIDIS) ha informado que desde el año 2016, los países latinoamericanos continúan empleando vertederos y botaderos de basura en el cielo abierto sin las

especificaciones técnicas, lo que ha generado la práctica de recolección sin clasificación. En un comparativo, solo el 17% de los desperdicios humanos se encuentran reciclados debido a lo previamente mencionado [Pachón, 2019].

Asimismo, las personas que llevan a cabo esta actividad de forma manual no están en las mejores condiciones de higiene y están expuestas a infecciones, lesiones o enfermedades. En general, la clasificación actual de estos desperdicios humanos provoca que se pierda materia prima y se pueda reintegrarse al ciclo de vida útil.

Este es un problema sumamente grave y es en este momento cuando surge la oportunidad de implementar modelos de visión por computadora para clasificar estos residuos y reducir dicha situación. [Pachón, 2019] elaboró un prototipo con inteligencia artificial y aprendizaje profundo, mediante una red neuronal convolucional, con el fin de clasificar empaques de plástico, vidrio y lata en el proceso de reciclaje, con un margen de error del 5%. En la componente de hardware, empleó Arduino para realizar la manipulación de una banda transportadora y una cámara que detecte los objetos. Asimismo, usó MATLAB para entrenar su algoritmo y crear una interfaz para la toma de decisiones del sistema. [Amaya, 2022] diseñó un sistema para detectar botellas de plástico y vidrio en los residuos mediante el uso del modelo YOLOv5, con una exactitud del 76% y una precisión del 86%. El sistema utilizó una banda transportadora y una cámara para detectar los tipos de botella. No obstante, está propenso a confundirse por algunos cambios de posición que podrían tener las botellas al momento de sujetarla. [Giraldo, 2022] empleó una red neuronal convolucional para detectar 12 grupos de residuos, tales como papel, cartón, lata, plástico, botellas plásticas, revistas, entre otros. Su base de datos fue obtenida a través de fotografías e internet, con un total de 3800 imágenes. En el entrenamiento obtuvo una precisión del 86% con imágenes no conocidas y el 98% con imágenes conocidas. El algoritmo fue ejecutado en tres sistemas distintos, tales como NVIDIA Jetson Nano, Dev Board Coral y Raspberry Pi. La velocidad de respuesta de su algoritmo en estos sistemas fue de entre 0.6 y 1.4 segundos. [Gómez, 2022] desarrolló un clasificador de residuos sólidos a través de una red neuronal. Su base de datos está compuesta por tres clases y entrenó su algoritmo con un total de 800 imágenes tomadas desde un dispositivo y

descargadas de internet. El algoritmo fue desarrollado a través de Google Teachable Machine, el código abierto de tensorflow y el editor P5.js, y obtuvieron una precisión del 89,63%. [Chen, 2020] planteó utilizar una mejora del modelo de YOLOv4 para detectar y clasificar 15 categorías de basura. La base de datos estaba compuesta por 22,000 imágenes para el conjunto de entrenamiento y 6,600 en el de prueba. Experimentó una comparativa de los modelos YOLOv3, YOLOv4 y la propuesta. Por consiguiente, el modelo que tuvo un rendimiento mejor fue la propuesta con un mAP del 64% y una velocidad de 92 FPS. [Ye, 2020] desarrolló un nuevo modelo de red neuronal de YOLO con el objetivo de acelerar el tiempo de procesamiento y precisión del reciclaje automatizado de basura. El rendimiento de su propuesta fue del 69.70%, alcanzando una velocidad de 60 FPS, lo que es más rápido que YOLOv1 y Fast R-CNN. Como se puede apreciar en la literatura, existe un gran interés en clasificar los desperdicios humanos mediante la utilización de versiones antiguas de YOLO, entre las cuales también se encuentran los trabajos de [De, 2020], [Wu, 2021].

En el presente artículo se propone implementar el modelo de visión por computadora YOLO en su versión 8 con el fin de clasificar diferentes tipos de residuos en tiempo real utilizando una cámara web. Este modelo, en comparación con las versiones de YOLO anteriores y utilizadas en el estado del arte (YOLOv3, YOLOv5 y YOLOv7), ha obtenido resultados con un mayor rendimiento en las aplicaciones en el área de transporte [Aboah, 2023] y monitoreo [Sharma, 2023]. La red neuronal convolucional de YOLOv8 ya ha sido usada previamente para la detección y clasificación de basura [Mudemfu, 2023], [Bawankule, 2023]. No obstante, en este estudio se centra en clasificar 5 categorías de los desperdicios humanos (metal, plástico, basura orgánica, papel y vidrio).

2. Métodos

Modelo de YOLOv8

En este trabajo se aplicó el modelo de YOLO, el cual fue creado por [Redmon, 2016] y publicado en *Computer Vision and Pattern Recognition*. El término YOLO significa “You Only Look Once” debido a su innovadora capacidad de detectar

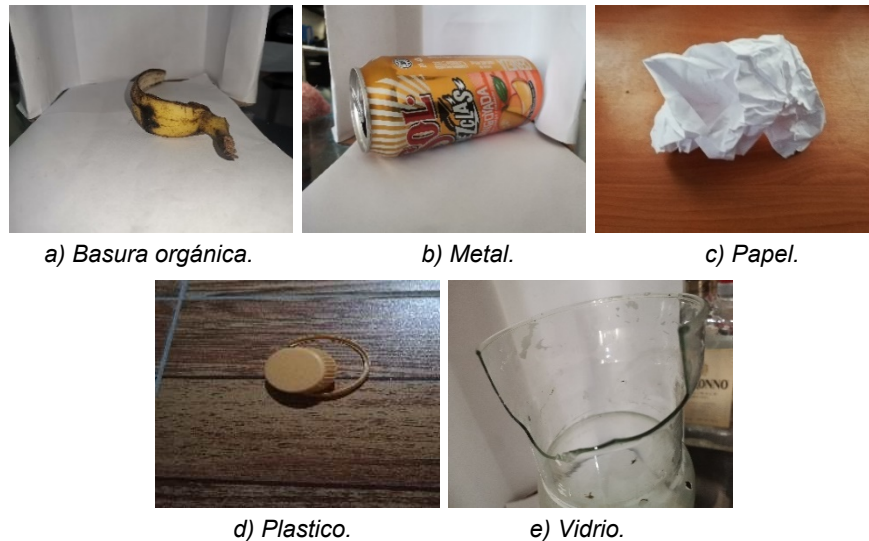
objetos en tiempo real con una sola pasada en la imagen. Esta técnica es diferente en comparación con los métodos anteriores, en los que se utilizaban ventanas deslizadoras que pasaban cientos de veces en la imagen para detectar la zona candidata [Vikram, 2017]. Asimismo, resulta menos robusto y difícil que aplicar las fases del procesamiento digital de imágenes [Vilet, 2005] combinado con técnicas tradicionales de aprendizaje automático. Las técnicas mencionadas anteriormente usan una gran cantidad de recursos computacionales, lo cual dificulta su implementación en tiempo real. Los modelos de YOLO son una red neuronal convolucional que extrae las características del objeto de interés en una sola pasada. Esto se realiza mediante el uso de capas convolucionales que transforman la imagen original en mapas de características. Durante el entrenamiento de este modelo, se comparan los valores de salida obtenidos de las neuronas de salida con el valor real, con el fin de ajustar los pesos W . Para evitar el sobreajuste de los pesos y mejorar la precisión, utiliza funciones de optimización para alcanzar los parámetros óptimos. Una de las funciones que determinan el rendimiento de estos modelos es loss function, la cual irá mejorando en cada una de las iteraciones [Gómez, 2022].

En la actualidad, este modelo ha ido mejorando con versiones nuevas. Una de las últimas modificaciones de YOLO es la versión YOLOv8, la cual fue concebida en enero del 2023 por Ultralytics [Terven, 2023], la misma compañía que desarrolló la versión de YOLOv5 [Jiang, 2022]. La red neuronal convolucional de YOLOv8 es similar a la de YOLOv5, pero tiene algunos cambios en algunas capas convolucionales. Esta red combina características de alto nivel y otros tipos de información para mejorar la precisión. Procesa las tareas de clasificación, regresión y estimación de poses de forma independiente. Este modelo utiliza ciertos elementos para calcular la pérdida y mejorar el rendimiento de la detección, especialmente de los objetos más pequeños. YOLOv8 dispone de 5 modelos de entrenamiento: YOLOv8n (nano), YOLOv8s (pequeño), YOLOv8m (mediano), YOLOv8l (largo) y YOLOv8x (muy largo). La principal diferencia entre los modelos es la cantidad de recursos computacionales necesarios y la precisión; cuanto más chico sea el modelo, más rápido será en la detección, pero su precisión disminuye.

A medida que el modelo sea más largo, tendrá más precisión, pero con más recursos requeridos.

Creación y preparación de las imágenes

Todo modelo de visión por computadora requiere una serie de datos para el entrenamiento, con el objetivo de extraer las características más relevantes del objeto de interés. La recolección de datos se llevó mediante la utilización de cinco cámaras de teléfono celular con una resolución de 4160 x 1968. Se seleccionaron cuidadosamente los objetos para la base en diversos ángulos de cámara y puntos de vista. La figura 1 presenta una parte del conjunto de datos.



Fuente: elaboración propia

Figura 1 Ejemplos del conjunto de datos.

Con esto en mente, se optó por realizar fotografías de los materiales más habituales en los contenedores de basura. El conjunto de datos está compuesto por cinco clases o categorías: papel, vidrio, metal, plástico y basura orgánica. El banco de datos se mejoró con un nuevo conjunto de imágenes de acceso libre del buscador de Google. Las imágenes tienen una resolución de 4160 x 1968 píxeles, y una profundidad de 32 bits (tres canales) de tipo .jpg. La base de datos se compone de 400 imágenes por clase o categoría, con el propósito de contar con un total de 2000 imágenes para el entrenamiento.

El entrenamiento del modelo requiere la imagen y las coordenadas de los objetos que se desean detectar en un archivo de formato .txt llamado "Label". El programa denominado "Labelimg" posibilita el etiquetado de objetos de interés en una imagen y la creación del archivo Label para el entrenamiento, figura 2.



Fuente: elaboración propia

Figura 2 Etiquetado del objeto en la imagen con Labelimg.

El archivo .txt que crea Labelimg muestra la información siguiente: 2 0.44555 0.33444 0.9333 0.223333. El primer número establece la categoría a la que pertenece el objeto, mientras que los otros valores establecen el tamaño y los puntos de localización del objeto de interés en la imagen. Posteriormente, la base de datos de imágenes y sus archivos .txt fueron divididos en tres conjuntos: entrenamiento, validación y el de prueba. El conjunto de entrenamiento se compone de imágenes que el modelo utilizará para entrenar y extraer las características de los objetos encontrados en las coordenadas de los Labels. El segundo grupo se usa para obtener el rendimiento del modelo durante cada una de las épocas de entrenamiento, lo cual es sumamente útil para guardar el modelo con mejor rendimiento. Finalmente, el conjunto de prueba se emplea para evaluar el rendimiento del modelo en imágenes que estuvieron fuera del entrenamiento. La literatura aconseja dividir la base de datos total en 80% para el conjunto de entrenamiento, 10% para el conjunto de prueba y 10% para el conjunto de validación [Landrum, 2004], [Muhammad, 2021].

Entrenamiento y métricas de medición

Para realizar el entrenamiento del modelo, se requiere la inclusión de imágenes del conjunto de entrenamiento, validación y los Labels. Los parámetros utilizados en este proceso se presentan en la tabla 1.

Tabla 1 Parámetros de entrenamiento.

Parámetro	Dato
Número de épocas	140
Batch	2
Modelo	YOLOv8l
Tamaño de imagen de entrada	640 x 640

El número de épocas empleadas en este trabajo fue de 140. El batch es el lote de imágenes que entran en la red neuronal antes de actualizar los pesos. Con el fin de alcanzar una precisión óptima y una mayor velocidad, se empleó el modelo de YOLOv8l. La resolución de la imagen de entrada en la capa convolucional se encuentra normalizada por defecto en un tamaño de 640 x 640. Existen una gran variedad de métricas para evaluar modelos de clasificación binaria y multiclase. Con YOLOv8 se pretende clasificar cinco categorías de desechos humanos, lo cual constituye un problema multiclase. La matriz de confusión en modelos de este tipo es mostrada en la tabla 2 [Markoulidakis, 2021].

Tabla 2 Matriz de aprendizaje multiclase.

		Categoría predicha			
		C_1	C_2	...	C_N
Categoría actual	C_1	$C_{1.1}$	FP	...	$C_{1.N}$
	C_2	FN	TP	...	FN

	C_N	$C_{N.1}$	FP	...	$C_{N.N}$

La matriz de confusión de la tabla 2 se compone de N filas por N columnas, donde N es el número de categorías. Se aplica la ecuación 1, para obtener la sensibilidad en un problema multiclase, la cual mide el porcentaje de objetos detectados que realmente pertenecen a una categoría.

$$TPR(C_i) = \frac{TP(C_i)}{TP(C_i) + FN(C_i)} \quad (1)$$

Donde:

$TPR(C_i)$: Sensibilidad de la clase i

$TP(C_i)$: Verdaderos positivos de la clase i

$FN(C_i)$: Falsos negativos de la clase i

En el caso de la precisión, se mide la proporción de objetos detectados que contienen realmente la categoría a clasificar. Esta métrica se calcula mediante la ecuación 2.

$$PPV(C_i) = \frac{TP(C_i)}{TP(C_i) + FP(C_i)} \quad (2)$$

Donde:

$PPV(C_i)$: Precisión de la clase i

$TP(C_i)$: Verdaderos positivos de la clase i

$FP(C_i)$: Falsos positivos de la clase i

Los verdaderos positivos son objetos detectados por el modelo y que en realidad pertenecen a la categoría que se encuentra. Los falsos positivos son objetos detectados, no obstante, no corresponden a la categoría real. En el caso de los falsos negativos, se trata de objetos que no detecto el modelo, pero se encontraban en la imagen. Para evaluar el rendimiento en la detección de objetos del modelo, es necesario incluir la precisión y sensibilidad en un valor único. La precisión media AP es el área debajo de la curva que se genera al integrar la precisión en función de la sensibilidad, la cual está definida por la ecuación 3 [Simón, 2020].

$$AP = \int_0^1 p(r) dr \quad (3)$$

Donde:

AP : Precisión media

$p(r)$: Precisión en función de la sensibilidad

Los límites son del 1 al 0, debido a que los porcentajes del AP varía entre esos valores. Durante el entrenamiento, se lleva a cabo la evaluación del modelo obtenido de esa época, junto con el conjunto de validación, con el fin de obtener las métricas previamente mencionadas. Para lograrlo, el AP utiliza un algoritmo denominado (IoU) y está representada por la ecuación número 4.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (4)$$

Donde:

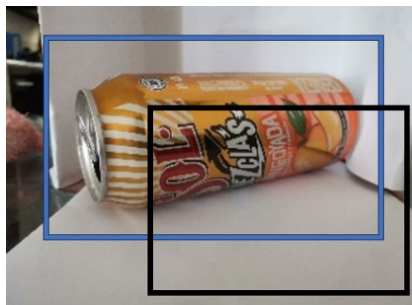
IoU : Intersección sobre la unión

$|A \cap B|$: Área de intersección

$|A \cup B|$: Área total de la unión

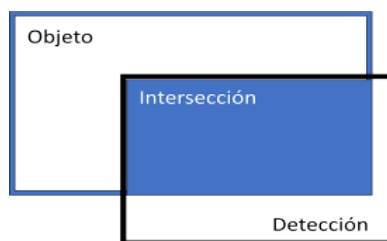
El IoU se utiliza para determinar si el objeto detectado pertenece a una clase real y es la relación entre el área de intersección y el área de unión del cuadro delimitador predicho. El conjunto de validación contiene imágenes previamente etiquetadas con Labelimg, las cuales tienen Labels con las coordenadas del objeto de interés. Este cuadro delimitador, enmarcado manualmente, se emplea para comparar con el cuadro delimitador detectado por el modelo.

La figura 3 presenta 2 cuadros marcados en el objeto de interés. El color azul representa el cuadro delimitador previamente marcado en el Labelimg, mientras que el negro representa el cuadro delimitador marcado por el modelo. El IoU para este ejemplo podría ser aproximadamente de 0.35, lo cual podría no considerarse una buena detección. En la figura 4, se muestra un ejemplo de la división de áreas.



Fuente: elaboración propia

Figura 3 Ejemplo de IoU de los dos cuadros delimitadores.



Fuente: elaboración propia

Figura 4 Intersección entre los dos cuadros delimitadores.

El *IoU* ayuda a determinar si la detección es un verdadero positivo o verdadero negativo. La toma de esta decisión se fundamenta en un umbral, y si el *IoU* es superior al umbral seleccionado, se considera un verdadero positivo. En caso contrario, la detección se considerará como un verdadero negativo. La métrica *mAP50* se refiere a la correlación entre la precisión y la sensibilidad del modelo mediante un umbral de 0.5 en el *IoU*, el cual fue empleado en este trabajo. Los resultados del mejor modelo durante el entrenamiento con el conjunto de validación se presentan en la tabla 3.

Tabla 3 Métricas obtenidas durante el entrenamiento.

Clase	Precisión	Sensibilidad	mAP50
Metal	0.831	0.799	0.878
Plástico	0.892	0.493	0.667
Basura orgánica	0.898	0.792	0.869
Papel	0.846	0.783	0.779
Vidrio	0.853	0.657	0.782
Total	0.864	0.705	0.795

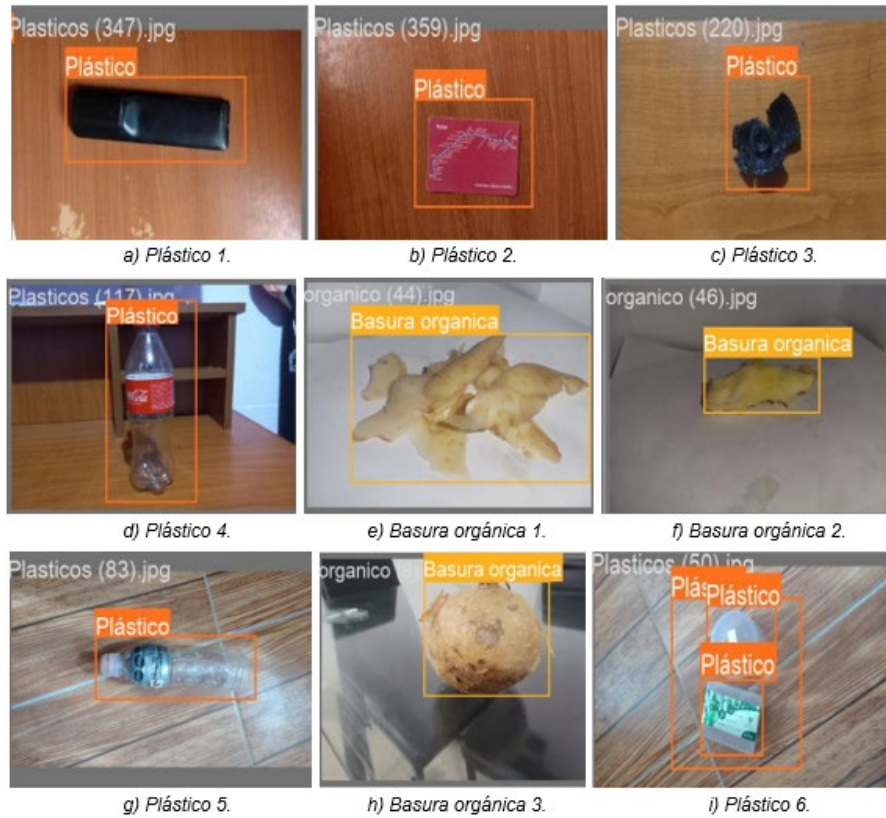
Fuente: elaboración propia

Las métricas obtenidas del modelo con mejor rendimiento durante las 140 iteraciones se presentan en la tabla número 3. La clase de plástico presenta dificultades en la clasificación debido a la presencia de aglomeraciones.

3. Resultados

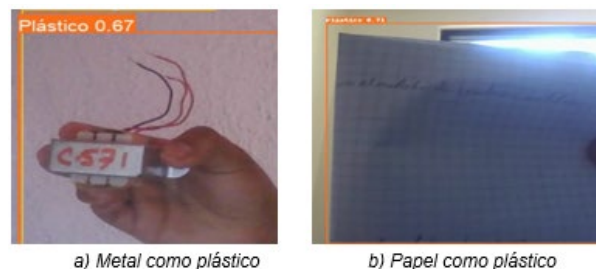
La primera revisión del modelo detectó los materiales correctamente y fue probado utilizando diferentes tipos de basura urbana.

Como se muestra en la figura 5, el modelo entrenado realizó algunas de las detecciones. No obstante, el modelo fue probado en tiempo real y se puede observar en la figura 6 que la clase metal y papel los confunde con otro objeto. Se llevó a cabo una segunda prueba con dos metales para efectuar una detección en la cámara web con este tipo de material. En la figura 7, se muestra que detectó correctamente los objetos de la clase metal a pesar del ruido y del desenfoque de la imagen. Al realizar estas detecciones con el conjunto de prueba, se obtuvieron los siguientes resultados.



Fuente: elaboración propia

Figura 5 Detecciones en imágenes del conjunto prueba.



Fuente: elaboración propia

Figura 6 Detecciones incorrectas.



Fuente: elaboración propia

Figura 7 Detecciones de la clase metal.

Los resultados de la tabla 4 muestran la precisión, la sensibilidad y el *mAP50*, con un conjunto de imágenes no utilizadas durante el entrenamiento. El modelo podría tener un fallo en la detección o reconocimiento incorrecto de ciertos objetos pertenecientes a la clase de papel y metal. La clasificación requiere un tiempo de 51 ms por fotograma para determinar y procesar la categoría del desecho humano.

Tabla 4 Resultados del conjunto prueba.

Clase	Precisión	Sensibilidad	mAP50
Metal	0.763	0.651	0.715
Plástico	0.874	0.641	0.826
Basura orgánica	0.869	0.827	0.896
Papel	0.894	0.619	0.747
Vidrio	0.922	0.556	0.695
Total	0.864	0.659	0.776

Fuente: elaboración propia

4. Discusión

El modelo puede detectar vidrio, metal, plástico, basura orgánica y papel con una precisión superior al 86%. No obstante, en ocasiones, se puede confundir el metal y el plástico. Se verificó que la clase plástico y vidrio es detectada correctamente y la clase metal en ocasiones se confunde con la categoría plástico. La proximidad a la cámara hacia el objeto de metal aumenta su clasificación óptima. Las categorías papel y basura orgánica tienen un menor rendimiento, ya que el modelo a veces detecta esos materiales como plástico o vidrio. Los problemas que surgieron se debieron a la posible presencia de fotografías desenfocadas y algunas de baja calidad obtenidas por la red. Este modelo fue implementado en tiempo real mediante la utilización de una cámara web para comprobar su velocidad y capacidad de detectar los objetos en el fotograma. La velocidad de procesamiento por fotograma es de 51 ms, lo cual significa que es un poco lento en clasificar los objetos, pero lo compensa con la precisión. Se propone como trabajo futuro entrenar el modelo con más imágenes en todas las categorías para aumentar la precisión, la sensibilidad y el *mAP50*. Asimismo, se incrementaría el número de épocas hasta 1000, con el fin reducir la cantidad de falsos positivos y detecciones incorrectas. Y, por último, usar un modelo más ligero como YOLOv8n o YOLOv8s para aumentar la velocidad de procesamiento en cada fotograma.

5. Conclusiones

La clasificación de los desperdicios humanos y su manipulación en vertederos no ha sido eficaz. Esto ha provocado que se generen concentraciones en términos de cantidad de toneladas por año de basura en los océanos y áreas terrestres. Los modelos de visión por computadora han demostrado resultados excelentes en la clasificación de residuos. En el presente estudio se ha implementado YOLOv8, un modelo de última generación para la detección y clasificación de objetos de tiempo real.

Este modelo fue entrenado con un total de 400 imágenes de la categoría papel, vidrio, metal, plástico y basura orgánica. Los resultados revelaron que la precisión del modelo de YOLOv8 en esta aplicación es superior al 86%, con una velocidad de inferencia de 50.2 ms por imagen. Por lo que, este modelo puede detectar los residuos en tiempo real utilizando una cámara web con una precisión considerablemente buena.

6. Bibliografía y Referencias

- [1] Aboah, A., Wang, B., Bagci, U., & Adu-Gyamfi, Y. Real-time multi-class helmet violation detection using few-shot data sampling technique and yolov8. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 5349-5357, 2023.
- [2] Amaya, L. D. E. Diseño e implementación de un prototipo para detección de envases en residuos sólidos por medio de visión artificial [Tesis doctoral, Universidad Santo Tomás]. Repositorio Institucional de la Universidad Santo Tomás <https://repository.usta.edu.co/handle/11634/48494>, 2022.
- [3] Bawankule, R., Gaikwad, V., Kulkarni, I., Kulkarni, S., Jadhav, A., & Ranjan, N. Visual Detection of Waste using YOLOv8. In 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS), 869-873, 2023.
- [4] Chen, Q., & Xiong, Q. Garbage classification detection based on improved YOLOv4. Journal of Computer and Communications, vol. 8, 285-294, 2020.

- [5] De, C. B., Ladogana, F., & Macchiarulo, N. Yolo trashnet: Garbage detection in video streams. In 2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), 1-7, 2020.
- [6] Giraldo, Q. J. D. Clasificador de residuos sólidos haciendo uso de deep learning. Universidad Autónoma de occidente, [https://red.uao.edu.co/bitstream/handle/10614/14209/T10373_Clasificador% 20de% 20residuos % 20sólidos% 20haciendo% 20uso% 20de% 20deep% 20learning](https://red.uao.edu.co/bitstream/handle/10614/14209/T10373_Clasificador%20de%20residuos%20sólidos%20haciendo%20uso%20de%20deep%20learning), 2022.
- [7] Gómez, V. D. L., & Tamayo, S. A. K. Clasificador de residuos sólidos para la ie Juan XXIII del municipio de Algeciras con aplicación de Machine Learning. *Revista Sennova: Revista del Sistema de Ciencia, Tecnología e Innovación*, 2022.
- [8] Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. A Review of Yolo algorithm developments. *Procedia Computer Science*, vol. 199, 1066-1073, 2022.
- [9] Landrum, G. A., Penzotti, J. E., & Putta, S. Machine-learning models for combinatorial catalyst discovery. *Measurement Science and Technology*, vol. 16, No. 1, 270, 2004.
- [10] Markoulidakis, I., Kopsiaftis, G., Rallis, I., & Georgoulas, I. Multi-class confusion matrix reduction method and its application on net promoter score classification problem. In *The 14th pervasive technologies related to assistive environments conference*, 412-419, 2021.
- [11] Mudemfu, M. K. Intelligent solid waste classification system using deep learning [Tesis Doctoral, Purdue University], https://hammer.purdue.edu/articles/thesis/INTELLIGENT_SOLID_WASTE_CLASSIFICATION_SYSTEM_USING_DEEP_LEARNING/23786484, 2023.
- [12] Muhammad, L. J., Algehyne, E. A., Usman, S. S., Ahmad, A., Chakraborty, C., & Mohammed, I. A. Supervised machine learning models for prediction of COVID-19 infection using epidemiology dataset. *SN computer science*, vol. 2, No. 1, 1-13, 2021.
- [13] Pachón, E. D. L. Prototipo de sistema automatizado con visión artificial para la selección de empaques de plástico, vidrio y lata en el proceso de reciclaje [Tesis de licenciatura, Universidad de Cundinamarca Extensión Chía].

- Repositorio Institucional de la Universidad de Cundinamarca Extensión Chía, <https://repositorio.ucundinamarca.edu.co/handle/20.500.12558/2497>, 2019.
- [14] Redmon, J., Divvala, S., Girshick R., & Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, 779-788, 2016.
- [15] Sharma, N., Baral, S., Paing, M. P., & Chawuthai, R. Parking Time Violation Tracking Using YOLOv8 and Tracking Algorithms. *Sensors*, vol. 23, No. 13, 5843, 2023.
- [16] Simón, A. R., & Gómez, R. R. Evaluación de algoritmos de machine learning para conducción [Trabajo Fin de Grado, Universidad Complutense Madrid], <https://docta.ucm.es/rest/api/core/bitstreams/bb1a8e13-13dd-4707-9658-973fc6a58fb1/content>, 2020.
- [17] Terven, J., Cordova-Esparza, D. A., & Romero-González, J. A. A comprehensive review of yolo architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-nas. *Machine Learning and Knowledge Extraction*, vol. 5, No. 4, 1680-1716, 2023.
- [18] Vilet, J. R. M. Procesamiento digital de imágenes. <http://read.pudn.com/downloads159/ebook/711796/ProcesamientoDigitaldelmagenes.pdf>, 2005.
- [19] Vikram, K., & Padmavathi, S. Facial parts detection using Viola Jones algorithm. In 2017 4th international conference on advanced computing and communication systems (ICACCS), 1-4, 2017.
- [20] Wu, Z., Zhang, D., Shao, Y., Zhang, X., Zhang, X., Feng, Y., & Cui, P. Using YOLOv5 for garbage classification. In 2021 4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), 35-38, 2021.
- [21] Ye, A., Pang, B., Jin, Y., & Cui, J. A YOLO-based neural network with VAE for intelligent garbage detection and classification. Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence, 1-7, 2020.