



Detection of Recyclable Solid Waste Using Convolutional Neural Networks and PyTorch

Jhandry R. Lapo , and Oscar M. Cumbicus-Pineda 

Abstract— Waste management in the recycling business is a time-consuming and labor-intensive process. In this context, the need to improve accuracy and reduce the time associated with this process is highlighted. In order to improve the classification of recyclable solid waste and streamline the waste management process, the creation of a convolutional neural network (CNN) model using PyTorch was proposed. The MLOps methodology was implemented in the development of the proposed model. In the first phase, an interview was conducted to analyze the waste sorting process in the company GIRA. In the second phase, the Taco Trash Dataset was reclassified, a CNN architecture based on RetinaNet was designed and the model was trained with hyper parameters based on related works. The third phase, the model was evaluated by testing and A/B testing. The model demonstrated high accuracy in waste detection and classification. It successfully identified materials such as paper, cardboard, PET bottles, hard plastic containers, flexible plastics, cans, glass, Tetra Pak containers, Flex foam and PET bottle caps. The loss was minimal, reaching 0.02120%, equivalent to 97% accuracy, and 80% accuracy in a real environment based on the Technology Acceptance Model (TAM). It is concluded that the implementation of a sorting and waste detection model optimizes the time and improves the accuracy of the sorting process.

Link to graphical and video abstracts, and to code: <https://latam.ieeer9.org/index.php/transactions/article/view/8764>

Index Terms— Computer vision, Image classification, Object detection, convolutional neural network, environment.

I. INTRODUCCIÓN

La gestión de los residuos sólidos constituye un desafío fundamental en la preservación del medio ambiente y la promoción de su sostenibilidad. Actualmente, dicha gestión se lleva a cabo mayoritariamente mediante procesos manuales, los cuales son notoriamente lentos, costosos y susceptibles a errores. Un factor determinante que contribuye a esta problemática es la carencia de infraestructura y tecnología idónea para la clasificación de residuos [1].

En este contexto, la aplicación de tecnologías avanzadas como el aprendizaje profundo y las redes neuronales convolucionales (CNN) surge como una alternativa prometedora para optimizar y perfeccionar el proceso de clasificación y detección de residuos sólidos reciclables [2], [3]. La presente investigación propone abordar esta necesidad

mediante el desarrollo de un modelo de detección y clasificación de residuos. La iniciativa se enfoca en la implementación de la librería PyTorch y CNN con el objetivo de mejorar tanto la eficiencia temporal como la precisión del proceso llevado a cabo manualmente por las empresas de reciclaje [4].

El principal objetivo planteado en esta investigación es "Crear un modelo de red neuronal convolucional para la detección y clasificación de residuos sólidos reciclables utilizando PyTorch". Para alcanzar este propósito, se han formulado dos fases. La primera fase de diseño, permitiendo así obtener un modelo entrenado a partir de datos reclasificados para cumplir con las diez clases utilizadas en el proceso de clasificación. La segunda fase de evaluación o testeo, con el propósito de evaluar la contribución positiva del modelo en la clasificación de los residuos.

Esta investigación fue estructurada en secciones que reflejan el proceso llevado a cabo para cumplir con los objetivos establecidos. Inicia por la Sección II de Antecedentes, seguidamente en la Sección III se muestra los Trabajos relacionados, luego en la Sección IV se abordan los materiales y métodos, en la Sección V se detallan los Resultados, a continuación, se presenta la sección VI de Discusión y finalmente se muestra la sección VII con las Conclusiones.

II. ANTECEDENTES

A. Detección de Residuos con Deep Learning (ML)

La detección de residuos mediante visión por computadora emplea algoritmos de aprendizaje automático para reconocer y clasificar diferentes tipos de residuos. Esta técnica implica el uso de cámaras y sensores para capturar imágenes de los residuos, seguido por el análisis de estas imágenes a través de software especializado que utiliza técnicas de visión por computadora y aprendizaje automático [5]. Los algoritmos, previamente entrenados con un conjunto de datos que incluye diversas imágenes de residuos, logran una detección y clasificación precisa [6].

B. Taco Trash Dataset

Conjunto de datos público de imágenes diseñado para entrenar y evaluar modelos de visión por computadora en la identificación de residuos o basura. Su objetivo principal es proporcionar una base de datos amplia y diversa para investigadores y desarrolladores, facilitando la mejora de algoritmos y modelos relacionados con la detección y segmentación de basura en imágenes [7]

III. TRABAJOS RELACIONADOS

Según la investigación realizada en trabajos relacionados con el tema de la presente investigación, como se muestra en la Tabla I, se concluyó que es posible clasificar residuos sólidos reciclables mediante el uso de tecnologías de aprendizaje profundo, como redes neuronales convolucionales y visión por computadora. Se observó que Python es el lenguaje de programación más utilizado debido a su abundancia de librerías de código abierto, su sintaxis simple y la disponibilidad de documentación extensa. Además, se optó por utilizar PyTorch junto con las tecnologías de Deep Learning mencionadas anteriormente.

TABLA I
TABLA DE TRABAJOS RELACIONADOS

ID	Trabajo relacionado	Herramienta	Tecnología	Ref.
TR01	Application of machine learning to waste management: identification and classification of recyclables.	Keras	CNN	[8]
TR02	Comparison of Re-trained CNN Models from PyTorch, Keras, and Tensorflow Frameworks for Image Waste Classification	PyTorch Keras Tensorflow	CNN	[9]
TR03	Application of Deep Learning for Solid Waste Trash Classification using Deep CNN	Pytorch	CNN	[10]
TR04	Clasificador de residuos sólidos haciendo uso de Deep Learning	Pytorch	CNN	[6]
TR05	Dry Waste Segregation using Computer Vision	Pytorch	CNN	[11]
TR06	Implementación de un sistema de detección de residuos reciclables basado en visión computacional	Pytorch	CNN	[12]
TR07	Deep Learning aplicado al procesamiento de imágenes para la detección de objetos reciclables	PyTorch, TensorFlow, Keras, OpenCV	CNN	[13]
TR08	Comparing deep learning and support vector machines for autonomous waste sorting	PyTorch	CNN SVM	[14]
TR09	Clasificador de residuos sólidos para la i.e. Juan XXIII del municipio de Algeciras con aplicación de Machine Learning	Tensorflow	CNN	[15]
TR10	Image based trash classification using machine learning	Tensorflow	CNN	[16]
TR11	algorithms for recyclability status Trash and Recycled Material Identification using Convolutional Neural Networks (CNN)	PyTorch	CNN	[17]

En el marco de esta investigación, es esencial analizar y comprender los enfoques y resultados obtenidos en trabajos previos relacionados con nuestro tema. Para ello, se compiló como se muestra en la Tabla II, que presenta de manera concisa los valores clave utilizados en estos trabajos, esta recopilación de información relevante, como tasas de aprendizaje (Learning Rate), número de épocas, entre otros, que han demostrado ser exitosos en investigaciones similares. Esta herramienta valiosa nos permitirá enriquecer nuestro enfoque y adaptar estrategias de manera informada para avanzar en la investigación.

TABLA II
VALORES DE REFERENCIA PARA PARÁMETROS DEL MODELO PROPUESTO

ID	Épocas	Op.	LR	Precisión	Perdida
TR01	N/A	N/A	N/A	0.92	N/A
1			A		
TR03	50	Adam	1e-3	0.97	N/A
TR04	30	Adam	1e-4	0.82	N/A
TR05	N/A	N/A	N/A	0.85	N/A
1			A		
TR06	N/A	Adam	N/A	0.53	N/A
1			A		
TR07	150	Adam	5e-3	0.74	N/A
1			A		
TR08	100	Adam	0.0	0.89	0.3
1			A		
TR09	50	N/A	1e-3	0.83	N/A
1			A		
TR10	60	N/A	0.0	0.89	N/A
1			A		
TR11	N/A	N/A	N/A	0.93	N/A
1			A		

IV. MATERIALES Y MÉTODOS.

En la presente investigación se desarrollaron 2 fases que abarcan el proceso de MLOps. En la primera fase de diseño y desarrollo de modelos, que se enfoca en comprender el contexto de la problemática. Esta fase se dividió en tres actividades (ver Sección V.A): (1) reclasificación de las clases basadas en la clasificación utilizada en la empresa GIRA de la ciudad de Loja para el entrenamiento del modelo, (2) diseño de la arquitectura del modelo y (3) El entrenamiento del modelo. En la segunda fase se aplicaron pruebas o test al modelo entrenado, esta fase se dividió en dos actividades (ver Sección V.B): (1) evaluación del modelo, misma que se realizó utilizando un conjunto de prueba o validación compuesto por imágenes no presentes en el conjunto de datos utilizado para entrenar el modelo y (2) se llevó a cabo un A/B testing comparando las predicciones del modelo con las clasificaciones proporcionadas por el responsable del proceso de clasificación de GIRA.

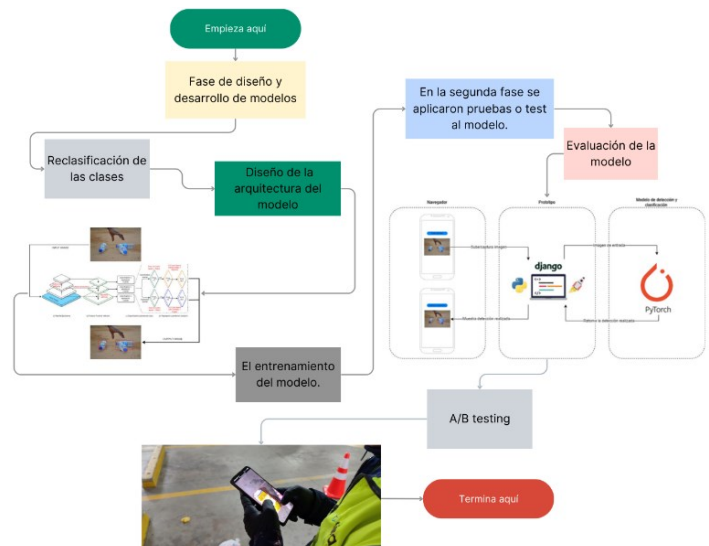


Fig. 1: Diagrama de materiales y métodos.

V. RESULTADOS

A. Fase I:

1) **Reclasificación del Dataset para la CNN**

Inicialmente se llevó a cabo una entrevista con el responsable del punto GIRA (usuario potencial) para obtener información sobre el proceso de recolección y clasificación de residuos. Un análisis de esta entrevista reveló las clases de residuos utilizadas en el punto GIRA como se muestra en la Tabla III, que sirven como base para el desarrollo del modelo, se determinó la necesidad de 10 clases de residuos, las cuales son utilizadas en el proceso de clasificación en dicha empresa.

TABLA III
CLASIFICACIÓN DE RESIDUOS EN LA EMPRESA GIRA

#	Clase
01	<i>Empaques PET</i>
02	<i>Tapas plásticas</i>
03	<i>Empaques de plástico flexible</i>
04	<i>Empaques de plástico rígido</i>
05	<i>Espumaflex (poliestireno)</i>
06	<i>Papel</i>
07	<i>Cartón</i>
08	<i>Tetra Pak</i>
09	<i>Vidrio</i>
10	<i>Latas</i>

Con el objetivo de obtener un dataset para el entrenamiento del modelo, fue esencial adquirir el conjunto de datos Taco Trash Dataset. Este dataset, diseñado para entrenar y evaluar modelos de detección de residuos, consta de 1580 imágenes junto con sus anotaciones. Estas anotaciones son las características que permiten al modelo de inteligencia artificial clasificar y detectar la ubicación de los residuos, asignándoles etiquetas correspondientes a las clases de residuos a las que pertenecen.

La Tabla IV presenta la reclasificación del conjunto de datos original (Taco Trash Dataset), que inicialmente tenía 60 clases y 15 superclases. Cada clase pertenece a una superclase del dataset, y se llevó a cabo una reducción a las 10 clases necesarias mediante la reclasificación de las 60 clases originales y su distribución en estas 10 clases.

TABLA IV
RECLASIFICACIÓN DE TACO TRASH DATASET A LAS 10 CLASES NECESARIAS PARA EL MODELO

Clase	S. O.	RC.
<i>Aluminium blister pack</i>	<i>Blister pack</i>	<i>Plásticos duros</i>
<i>Carded blister pack</i>	<i>Blister pack</i>	<i>Plásticos duros</i>
<i>Other plastic bottle</i>	<i>Bottle</i>	<i>Empaques PET</i>
<i>Clear plastic bottle</i>	<i>Bottle</i>	<i>Empaques PET</i>
<i>Glass bottle</i>	<i>Bottle</i>	<i>Vidrio</i>
<i>Plastic bottle cap</i>	<i>Bottle cap</i>	<i>Tapas plásticas</i>
<i>Broken glass</i>	<i>Broken glass</i>	<i>Vidrio</i>
<i>Food Can</i>	<i>Can</i>	<i>Lata</i>
<i>Aerosol</i>	<i>Can</i>	<i>Lata</i>
<i>Drink can</i>	<i>Can</i>	<i>Lata</i>
<i>Toilet tube</i>	<i>Carton</i>	<i>Cartón</i>

<i>Other carton</i>	<i>Carton</i>	<i>Cartón</i>
<i>Egg carton</i>	<i>Carton</i>	<i>Cartón</i>
<i>Drink carton</i>	<i>Carton</i>	<i>Envases de Tetra Pak</i>
<i>Corrugated carton</i>	<i>Carton</i>	<i>Cartón</i>
<i>Meal carton</i>	<i>Carton</i>	<i>Cartón</i>
<i>Pizza box</i>	<i>Carton</i>	<i>Cartón</i>
<i>Paper cup</i>	<i>Cup</i>	<i>Plásticos flexibles</i>
<i>Disposable plastic cup</i>	<i>Cup</i>	<i>Plásticos flexibles</i>
<i>Foam cup</i>	<i>Cup</i>	<i>Espuma Flex</i>
<i>Glass cup</i>	<i>Cup</i>	<i>Vidrio</i>
<i>Other plastic cup</i>	<i>Cup</i>	<i>Plásticos duros</i>
<i>Glass jar</i>	<i>Glass jar</i>	<i>Vidrio</i>
<i>Plastic lid</i>	<i>Lid</i>	<i>Plásticos duros</i>
<i>Other plastic</i>	<i>Other plastic</i>	<i>Plásticos duros</i>
<i>Magazine paper</i>	<i>Paper</i>	<i>Papel</i>
<i>Wrapping paper</i>	<i>Paper</i>	<i>Papel</i>
<i>Normal paper</i>	<i>Paper</i>	<i>Papel</i>
<i>Paper bag</i>	<i>Paper bag</i>	<i>Plásticos flexibles</i>
<i>Plastified paper bag</i>	<i>Paper bag</i>	<i>Papel</i>
<i>Plastic film</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos duros</i>
<i>Six pack rings</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos flexibles</i>
<i>Garbage bag</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos flexibles</i>
<i>Other plastic wrapper</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos flexibles</i>
<i>Single-use carrier bag</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos flexibles</i>
<i>Polypropylene bag</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos flexibles</i>
<i>Crisp packet</i>	<i>Plastic bag & wrapper</i>	<i>Plásticos flexibles</i>
<i>Spread tub</i>	<i>Plastic container</i>	<i>Plásticos duros</i>
<i>Tupperware</i>	<i>Plastic container</i>	<i>Plásticos duros</i>
<i>Disposable food container</i>	<i>Plastic container</i>	<i>Empaques PET</i>
<i>Foam food container</i>	<i>Plastic container</i>	<i>Espuma Flex</i>
<i>Other plastic container</i>	<i>Plastic container</i>	<i>Plásticos duros</i>
<i>Plastic glooves</i>	<i>Plastic glooves</i>	<i>Plásticos flexibles</i>
<i>Plastic utensils</i>	<i>Plastic utensils</i>	<i>Plásticos duros</i>
<i>Plastic straw</i>	<i>Straw</i>	<i>Plásticos flexibles</i>
<i>Paper straw</i>	<i>Straw</i>	<i>Papel</i>
<i>Styrofoam piece</i>	<i>Styrofoam piece</i>	<i>Espuma Flex</i>
<i>Aluminium foil</i>	<i>Aluminium foil</i>	<i>Desconocido</i>
<i>Battery</i>	<i>Battery</i>	<i>Batería</i>
<i>Metal bottle cap</i>	<i>Bottle cap</i>	<i>Desconocido</i>
<i>Food waste</i>	<i>Food waste</i>	<i>Inorgánico</i>
<i>Metal lid</i>	<i>Lid</i>	<i>Desconocido</i>
<i>Tissues</i>	<i>Paper</i>	<i>Desconocido</i>
<i>Pop tab</i>	<i>Pop tab</i>	<i>Desconocido</i>
<i>Rope & strings</i>	<i>Rope & strings</i>	<i>Desconocido</i>
<i>Scrap metal</i>	<i>Scrap metal</i>	<i>Desconocido</i>
<i>Shoe</i>	<i>Shoe</i>	<i>Desconocido</i>
<i>Squeezable tube</i>	<i>Styrofoam piece</i>	<i>Desconocido</i>
<i>Unlabeled litter</i>	<i>Unlabeled litter</i>	<i>Desconocido</i>
<i>Cigarette</i>	<i>Cigarette</i>	<i>Desconocido</i>

Superclases Originales (S. O.); Reclasificación (RC)

La reclasificación se llevó a cabo mediante un script de Python, ejecutado en el servicio de Google Colab. Como resultado se muestra las nuevas clases asignadas, logrado mediante el uso de un diccionario que permitió asignar nuevos valores de superclases en las anotaciones del conjunto de datos original. Estas anotaciones fueron guardadas como un objeto .json que se utilizó para el entrenamiento del modelo.

2) **Diseñar la Arquitectura de la CNN para la Detección de Residuos Sólidos Reciclable**

Para el diseño la arquitectura de la Red Neuronal Convolutiva (CNN) para el entrenamiento del modelo de detección de residuos, se tomó como base la estructura de RetinaNet (desarrollado por Facebook AI Research en 2018, basándose en la arquitectura ResNet 18, 34, 50, 101, 152 [18][19]). En la presente investigación, se propuso un modelo utilizando ResNet-50 (es una red neuronal residual desarrollada por Microsoft Research en 2015 [20]).

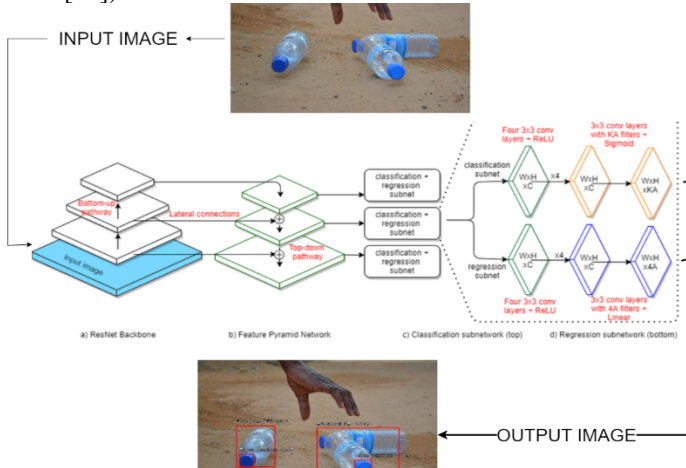


Fig. 2: Arquitectura del modelo de detección de residuos basada en RetinaNet.

La arquitectura que se diseñó consta de dos componentes: el armazón (backbone) y el head. El armazón es una combinación de una rama ascendente, que es una red residual (ResNet-50 en nuestro caso), y una rama descendente, que es una red piramidal de características (Feature Pyramid Network, FPN). La rama descendente extrae características en diferentes capas de la arquitectura para abordar el desafío de la detección de objetos pequeños o de diferentes tamaños en las imágenes. La rama ascendente mapea las características de la imagen de entrada y las reduce a la mitad, transfiriendo información a la capa equivalente de la rama descendente. Por otro lado, la rama descendente duplica la dimensión del mapa de características de capas anteriores. En el head, hay dos subredes: una de clasificación, que realiza predicciones de las clases de los objetos en la imagen (por ejemplo, cartón, papel, entre otros, en el caso del estudio actual), y una subred de regresión, que predice la ubicación de los objetos en la imagen utilizando regresión. El head extrae información de las salidas de cada capa de la rama descendente, aplicando las subredes de clasificación y regresión en cada una de estas capas.

3) **Entrenar el Modelo Utilizando el Conjunto de Datos**

Para este punto se estableció los DataFrames o conjuntos de datos que contenían las imágenes y sus anotaciones, separados en conjuntos de entrenamiento y validación. La definición de estos DataFrames se llevó a cabo mediante la herramienta Google Colab [4] y la librería Pandas. Este script realizó una división que asignó como se puede ver en la Fig. 3 el 5% de los datos al conjunto de validación,

mientras que el 95% restante se destinó al conjunto de entrenamiento, esta división se considera debido a que se realizó pruebas A/B testing en la siguiente fase. En otras palabras, el 95% de los datos originales, equivalentes a 1501 imágenes con sus anotaciones, se emplearon para entrenar el modelo de aprendizaje automático. Por otro lado, el 5%, conformado por 79 imágenes restantes, se reservó para evaluar el rendimiento del modelo entrenado. Esta división resulta crucial para evaluar la capacidad del modelo en datos no vistos durante el entrenamiento, utilizando así el DataFrame de validación para evitar sesgos.

División de conjunto de entrenamiento y validación

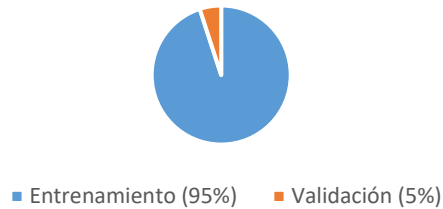


Fig. 3: Conjunto de validación y entrenamiento.

Después de definir los conjuntos de datos de entrenamiento y validación, como se muestra en la Tabla VI, es necesario establecer los DataLoader y los hiperparámetros para el entrenamiento y validación del modelo. Se seleccionó un batch_size de 2, y para la evaluación del modelo, se definió un valor de 1, lo que implica que cada muestra se procesará individualmente. En ambos DataLoaders, el parámetro num_workers, que indica los subprocesos para cargar los datos utilizando varios núcleos del procesador para acelerar el proceso, se estableció en 3. Además, para el entrenamiento, se definió el optimizador Adam, ampliamente utilizado y efectivo en trabajos relacionados, junto con otros hiperparámetros, como se muestra en la Tabla V.

TABLA V
HIPER PARÁMETROS PARA EL CONJUNTO DE ENTRENAMIENTO Y VALIDACIÓN

DL	Épocas	Batch Size	NW	LR
<i>Train</i>	100	2	3	1e-5
<i>Val</i>	<i>N/A</i>	1	3	<i>N/A</i>

Num_workers (NW), Learning Rate (LR)

Para el entrenamiento inicial del modelo, se intentó realizar una prueba de entrenamiento de manera local en una computadora portátil con una GPU de 6 GB de memoria. Sin embargo, las capacidades computacionales de esta máquina no fueron suficientes para cumplir con los requisitos necesarios para el entrenamiento del modelo. En la Tabla VI se presentan los parámetros que se utilizaron para esta prueba.

TABLA VI

HIPER PARÁMETROS DEFINIDOS PARA EL ENTRENAMIENTO DE PRUEBA

#	Épocas	Batch Size	LR
1	6	6	1e-5
2	5	4	1e-5
3	4	2	1e-5

Learning Rate (LR)

Para llevar a cabo los entrenamientos del modelo de prueba, se eligió utilizar la plataforma de Google Colab. Esta herramienta permite acceder de forma gratuita a una GPU (T4) con aproximadamente 15 GB de memoria durante un tiempo limitado. De esta manera, se logró completar los primeros entrenamientos del modelo utilizando los parámetros de la TABLA VI. Sin embargo, como se muestra en la Fig. 4, los valores de pérdida fueron considerablemente altos, lo que indica que los modelos de prueba entrenados no realizaron predicciones acertadas.

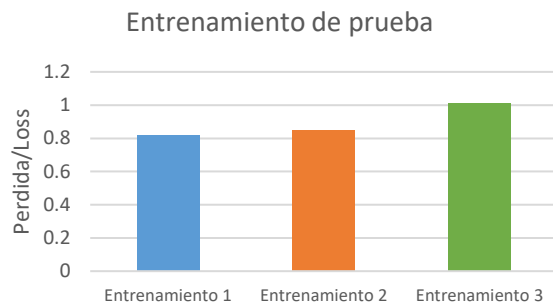


Fig. 4: Entrenamientos de prueba.

Después de evaluar los porcentajes de precisión y pérdida obtenidos en los entrenamientos del modelo de prueba, se decidió aumentar el número de épocas a 30. Esta elección se hizo debido a restricciones diarias de uso de las GPU proporcionadas por Google, así como desconexiones por inactividad. En la Tabla VII, se presentan los valores de pérdida (loss) obtenidos en las últimas 5 iteraciones del entrenamiento del modelo. Estos valores guardan una relación inversamente proporcional con la precisión del modelo, aunque los valores de pérdida disminuyeron y, en consecuencia, la precisión aumentó en comparación con el modelo de prueba, esta sigue siendo baja, por lo tanto, se concluye que es necesario aumentar el número de épocas en las que se realiza el entrenamiento del modelo.

TABLA VII

VALOR DE PÉRDIDA DEL MODELO ENTRENADO CON 30 ÉPOCAS

Iteración	Perdida/Loss
505	0.25590
510	0.25586
515	0.25569
520	0.25541
525	0.25511

Para aumentar el número de épocas y reducir la pérdida del modelo, era necesario mantener activo Google Colab durante más tiempo, lo cual representaba un problema debido al límite de tiempo de la versión gratuita de la plataforma. Para superar esta limitación, se decidió adquirir Google Colab Pro. Esta

versión no solo permitió superar la restricción de tiempo, sino que también proporcionó acceso a una GPU A100 de 40 GB, mejorando significativamente las capacidades computacionales [4]. Sin embargo, estas aún no fueron suficientes para completar el entrenamiento con las épocas necesarias. Ante esta situación, se exploró la alternativa de solicitar acceso a la plataforma de ordenador de alto rendimiento (HPC) de la empresa CEDIA. CEDIA ofrece acceso a esta plataforma para proyectos que requieren este servicio, y se realiza la solicitud enviando un correo electrónico al servicio NOC, conforme a las instrucciones proporcionadas en su página web [4]. Después de solicitar el acceso, se obtuvo una respuesta positiva por parte de CEDIA, se logró acceder a la plataforma de HPC, posteriormente se configuró la plataforma y luego de aproximadamente 23 horas de entrenamiento, se completaron las 100 iteraciones o épocas como se puede ver en la Fig.5, mismas que fueron propuestas en la TABLA VII de los hiperparámetros para el modelo final. El resultado fue una pérdida o loss de 0.02120 en el entrenamiento, lo que indica que se logró una precisión de alrededor del 97%.

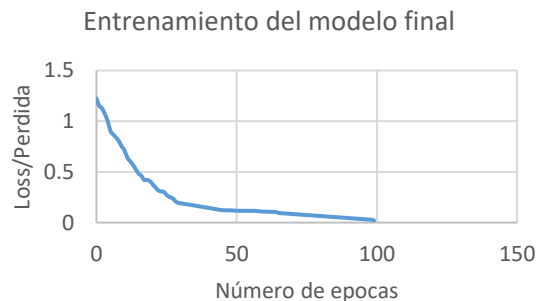


Fig. 5: Culminación del proceso de entrenamiento del modelo.

B. Fase 2

1. Evaluar el modelo

Una vez completado el entrenamiento del modelo, es esencial realizar una evaluación para validar el valor de pérdida o loss obtenido. Para evitar sesgos en la validación o evaluación del rendimiento del modelo, se realizaron pruebas con imágenes que el modelo no había utilizado durante el proceso de entrenamiento. Estas imágenes representan diversos escenarios y objetos, lo que permite evaluar la capacidad del modelo para detectar y clasificar residuos sólidos reciclables. En la Fig. 6 se muestran los resultados de las pruebas llevadas a cabo en el modelo [4].



Fig. 6: Prueba de detección realizada a imagen.

En la Tabla VIII se detallan parte de las evaluaciones llevadas a cabo. Esta evaluación se ejecutó utilizando el 5% del conjunto de datos, que equivale a 79 imágenes destinadas específicamente para esta tarea.

TABLA VIII
PRUEBAS REALIZADAS AL MODELO ENTRENADO

#	Clasificación	F.C	Resultado obtenido
1	<i>Empaques PET</i>	0.2	<i>Empaques PET</i>
2	<i>Plástico flexible</i>	0.2	<i>Plástico flexible</i>
3	<i>Empaques PET</i>	0.2	<i>Empaques PET</i>
4	<i>Latas</i>	0.2	<i>Latas</i>
5	<i>Latas</i>	0.2	<i>Latas</i>
6	<i>Plástico duro</i>	0.2	<i>Plástico duro</i>
7	<i>Empaques PET</i>	0.2	<i>Empaques PET</i>
8	<i>Latas</i>	0.2	<i>Vidrio</i>
9	<i>Empaques PET</i>	0.2	<i>Empaques PET</i>
10	<i>Plástico flexible</i>	0.2	<i>Plástico flexible</i>
11	<i>Empaques PET</i>	0.2	<i>Empaques PET</i>
12	<i>Plástico flexible</i>	0.2	<i>Plástico flexible</i>
13	<i>Empaques PET</i>	0.2	<i>Empaques PET</i>
14	<i>Plástico flexible</i>	0.2	<i>Plástico flexible</i>
15	<i>Latas</i>	0.2	<i>Latas</i>

Factor de confianza (F.C)

La Fig.7 presenta la matriz de confusión, la cual se utilizó para visualizar los resultados de las 100 pruebas realizadas. Esta matriz nos ofrece una representación gráfica de la precisión del modelo en cada una de las 10 clases de residuos que se utilizaron en el presente estudio.

		MATRIZ DE CONFUSIÓN										
TARGET \ OUTPUT	EMPAQUES PET	TAPAS PLÁSTICAS	PLÁSTICO FLEXIBLE	PLÁSTICO DURO	ESPUMA FLEX	PAPEL	CARTÓN	TETRA PACK	VIDRIO	LATAS	SUM	
EMPAQUES PET	24 24.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	03 8.88%	27 88.88% 11.11%
TAPAS PLÁSTICA	0 0.00%	03 3.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	3 100.00% 0.00%
PLÁSTICO FLEXIB	0 0.00%	0 0.00%	25 25.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	1 1.00%	0 0.00%	26 96.15%	26 100.00% 3.85%
PLÁSTICO DURO	0 0.00%	0 0.00%	0 0.00%	7 7.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	7 100.00%	7 100.00% 0.00%
ESPUMA FLEX	0 0.00%	0 0.00%	0 0.00%	0 0.00%	5 5.00%	1 1.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	6 83.33%	6 100.00% 16.67%
PAPEL	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	4 4.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	4 100.00%	4 100.00% 0.00%
CARTÓN	0 0.00%	0 0.00%	0 0.00%	1 1.00%	0 0.00%	0 0.00%	3 3.00%	0 0.00%	0 0.00%	0 0.00%	4 75.00%	4 100.00% 25.00%
TETRA PACK	0 0.00%	0 0.00%	1 1.00%	0 0.00%	0 0.00%	0 0.00%	4 4.00%	0 0.00%	0 0.00%	0 0.00%	5 80.00%	5 100.00% 20.00%
VIDRIO	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	8 8.00%	0 0.00%	0 0.00%	8 100.00%	8 100.00% 0.00%
LATAS	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	0 0.00%	1 1.00%	9 9.00%	10 90.00%	10 100.00% 10.00%
SUM	24 100.00%	3 100.00%	26 96.15%	7 87.50%	5 100.00%	4 80.00%	3 100.00%	4 100.00%	10 88.00%	12 75.00%	92 / 100 92.00%	100 100.00%

Fig. 7: Matriz de confusión de la evaluación realizada al modelo.

Por otro lado, la Tabla IX presenta los resultados de la matriz de confusión, donde se logró una precisión del 0.92 en la evaluación realizada. Este resultado muestra una diferencia con la validación llevada a cabo durante el entrenamiento del modelo, donde la precisión, por deducción, fue del 0.97.

TABLA IX
RESULTADOS DE LA MATRIZ DE CONFUSIÓN

#	Clase	Pre.	1-P	Recall	1-Recall	F1-Score
1	<i>Empaques PET</i>	0,8889	0,1111	1,0000	0,0000	0,9412
2	<i>Tapas plásticas</i>	1,0000	0,0000	1,0000	0,0000	1,0000
3	<i>Plástico flexible</i>	0,9615	0,0385	0,9615	0,0385	0,9615
4	<i>Plástico duro</i>	1,0000	0,0000	0,8750	0,1250	0,9333
5	<i>Espuma Flex</i>	0,8333	0,1667	1,0000	0,0000	0,9091
6	<i>Papel</i>	1,0000	0,0000	0,8000	0,2000	0,8889
7	<i>Cartón</i>	0,7500	0,2500	1,0000	0,0000	0,8571
8	<i>Tetra Pak</i>	0,8000	0,2000	1,0000	0,0000	0,8889
9	<i>Vidrio</i>	1,0000	0,0000	0,8000	0,2000	0,8889
10	<i>Latas</i>	0,9000	0,1000	0,7500	0,2500	0,8182

2. Evaluar el modelo mediante A/B Testing

Para llevar a cabo las pruebas de A/B testing propuestas en esta tarea, se desarrolló un prototipo de plataforma web utilizando el framework Django, como se muestra en la Fig. 8, que facilitó la utilización del modelo de red neuronal convolucional previamente entrenado y simplificar el proceso de realizar pruebas.



Fig. 8: Prototipo web desarrollado para utilizar el modelo obtenido en el objetivo 1.

En la Fig. 9, se muestra la arquitectura del prototipo que permite capturar o cargar imágenes para realizar pruebas y obtener las predicciones del modelo mediante el uso de dispositivos como teléfonos móviles o computadoras que dispongan de cámara web.

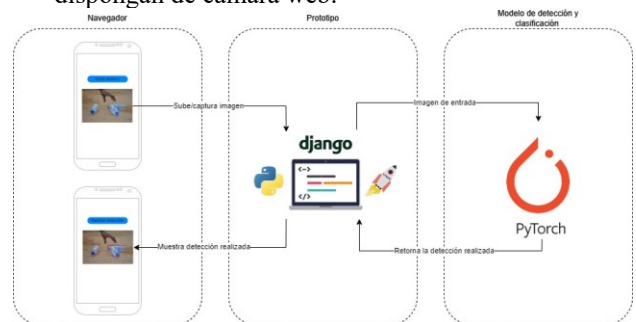


Fig. 9: Arquitectura de prototipo web de detección y clasificación de residuos.

Una vez finalizado el prototipo, se realizó la evaluación mediante A/B testing. Este proceso implicó la comparación de la clasificación realizada por el modelo entrenado utilizando la plataforma web como sujeto A,

con la clasificación tradicional realizada por el encargado del punto GIRA en el proceso interno de la empresa como sujeto B. Se registraron los resultados de las predicciones realizadas por ambos sujetos y se compararon sus desempeños utilizando métricas específicas, como la precisión de detección en el caso del modelo y el tiempo que les tomó realizar el proceso con el mismo número de residuos. Este proceso se llevó a cabo en 5 pruebas, donde se realizaron comparativas entre los resultados de los sujetos [4].

Prueba 1: En la primera prueba se utilizaron 5 objetos, como se puede observar en la Fig. 10, en la Tabla X se muestra la comparativa de los resultados obtenidos de los 2 sujetos de prueba.

Análisis: El modelo acertó 5 de los 5 residuos de la prueba y el experto acertó 5 de los 5 residuos de la prueba. El tiempo que requirió el modelo para realizar las predicciones de los 5 objetos fue de 23 segundos y al experto le tomó 30 segundos. Aunque el tiempo mejoró por una diferencia de 7 segundos, el modelo realizó predicciones incorrectas.

TABLA X
RESULTADOS DE LA PRIMERA PRUEBA

Obj.	Clase R.	P. M.	Pr. (%)	P. E.
01	<i>Empaques PET</i>	<i>Empaques PET</i>	18	<i>Empaques PET</i>
02	<i>Tapas plásticas</i>	<i>Tapas plásticas</i>	7	<i>Tapas plásticas</i>
03	<i>Envases de Tetra Pack</i>	<i>Envases de Tetra Pack</i>	8	<i>Envases de Tetra Pack</i>
04	<i>Latas</i>	<i>Latas</i>	73	<i>Latas</i>
05	<i>Latas</i>	<i>Latas</i>	79	<i>Latas</i>



Fig. 10: Predicciones realizadas en la primera prueba.

En la evaluación del modelo de detección y clasificación de residuos, el experto logró una precisión del 100%, acertando en todos los 34 residuos probados, demostrando su experiencia. Por otro lado, el modelo tuvo una precisión del 67.64%, acertando en 23 de los 34 residuos. El tiempo promedio de clasificación fue de 35 segundos para el experto, variando según la cantidad de objetos, mientras que el modelo mantuvo un promedio de 23 segundos, independientemente de la cantidad de objetos en la imagen. Este tiempo está sujeto a las

capacidades computacionales del dispositivo; en una prueba adicional en un dispositivo más potente, el tiempo se redujo a alrededor de 2 segundos, pero cabe aclarar que esta reducción de tiempo de cerca del 80% no mejora la precisión del modelo, por lo que, aunque el tiempo se ve reducido, la precisión sigue teniendo un valor considerablemente bajo.

Una vez culminadas las pruebas de A/B testing para evaluar el modelo, se utilizó el Modelo de Aceptación Tecnológica (TAM) para medir la aceptabilidad del prototipo de detección y clasificación de residuos en el punto GIRA [4]. El participante estuvo de acuerdo en un 80% con la influencia positiva del modelo en la precisión de la clasificación y detección de residuos, y en un 100% en que el modelo aportaría significativamente en la reducción del tiempo actualmente requerido para el proceso de clasificación en el punto GIRA. Además, se realizaron preguntas sobre la facilidad de uso, intención de uso y mejoras necesarias para el correcto funcionamiento del prototipo.

VI. DISCUSIÓN

Para la primera fase de la investigación, se llevaron a cabo las actividades para el desarrollo del modelo de detección de residuos. Aunque existen diversas técnicas que podrían haber contribuido a culminar este objetivo, se realizó un análisis de los trabajos relacionados del TR01 al TR11 en los que se observó que estos trabajos utilizan conjuntos de imágenes diferentes y no proporcionan acceso a los mismos. Por ende, en este objeto de estudio, fue necesario obtener un conjunto de datos para entrenar el modelo propuesto. En cuanto a las tecnologías utilizadas, la mayoría de los trabajos relacionados (TR01 al TR11) se basan en el uso de Redes Neuronales Convolucionales (CNN). En el diseño de la arquitectura de la red para el modelo, se adoptó este enfoque, ya que está diseñado principalmente para la detección de objetos en imágenes o videos. Se eligió la librería PyTorch, siguiendo la elección de TR02, TR03, TR04, TR05, TR08 y TR11, debido a la experiencia académica del estudiante autor de la presente investigación. En relación con los hiperparámetros, se hizo un análisis detallado de su relevancia en la arquitectura del modelo. Se registraron los valores utilizados en cada uno de los trabajos relacionados del TR01 al TR11 en una tabla (TABLA III). Se definió un número de épocas de 100, tomando como base TR08 para así evitar el sobreentrenamiento. Con estos pasos, se logró entrenar el modelo de detección y clasificación propuesto. Sin embargo, se destaca la importancia de considerar la diversidad de conjuntos de datos y la elección adecuada de tecnologías según la experiencia y contexto específico del proyecto.

Para la segunda fase de evaluación el modelo entrenado, se utilizó un conjunto de validación constituido por el 5% del total de datos, es decir, 79 imágenes, es decir, que utiliza la mayoría de datos para el entrenamiento y un porcentaje pequeño para la validación debido al tamaño del conjunto de datos y a las pruebas adicionales que se realizó. A diferencia de estos, se seleccionaron imágenes completamente distintas al conjunto de entrenamiento para evitar sesgos. Contrariamente a los trabajos relacionados, se incorporó una evaluación adicional basada en

la metodología MLOps. Esta evaluación simula un entorno real similar al proceso de clasificación de residuos en el punto GIRA. Aunque la primera evaluación confirmó una precisión del 97% en el entrenamiento, la segunda evaluación reveló un porcentaje de precisión sustancialmente menor, alcanzando el 67%. Esta diferencia se atribuye a factores como el entorno de las imágenes en el dataset y las librerías utilizadas en el modelo. Es esencial destacar que, a diferencia de los trabajos relacionados que emplean librerías especializadas como TensorFlow o Yolo para la detección de objetos, la elección de PyTorch en esta investigación podría haber afectado la precisión del modelo en un entorno real. Estos hallazgos proporcionan una comprensión más profunda del rendimiento del modelo en situaciones prácticas y subrayan la importancia de considerar la idoneidad de las herramientas y librerías utilizadas en el desarrollo de modelos de detección de residuos.

Finalmente, los resultados obtenidos a través del Modelo de Aceptación Tecnológica indican una aceptación del 100% en cuanto a la mejora del tiempo en el proceso de clasificación de la empresa GIRA, y un 80% de aceptación en la precisión de las predicciones generadas por el modelo. Estos hallazgos sugieren que la implementación de un modelo de redes neuronales convolucionales con PyTorch puede tener un impacto positivo en la eficiencia del proceso de clasificación de residuos en la empresa GIRA. La aceptación del 80% en la precisión de las predicciones respalda la utilidad y viabilidad del modelo desarrollado. Sin embargo, es importante tener en cuenta las limitaciones identificadas durante las evaluaciones, como la variabilidad en la precisión en un entorno real en comparación con el entorno de entrenamiento.

VII. CONCLUSIONES

La creación y aplicación del modelo de red neuronal convolucional en PyTorch para la detección y clasificación de residuos sólidos demostró ser una herramienta valiosa en el punto GIRA de la ciudad de Loja. Este enfoque ha demostrado reducir significativamente el tiempo necesario para el proceso de clasificación, al tiempo que mejora la precisión en un 80%, lo que potencialmente contribuirá a una gestión más eficiente de los residuos y promoverá prácticas sostenibles en el reciclaje. Se destaca que los hiperparámetros utilizados en el entrenamiento del modelo no son constantes y su eficacia depende de varios factores, incluyendo el conjunto de datos y el diseño de la arquitectura de red. Se enfatiza la importancia de experimentar con valores como el learning rate, el número de épocas y las funciones de activación para adaptarse a las necesidades específicas de la investigación.

La implementación de la metodología MLOps fue fundamental, ya que aborda de manera integral cada fase del desarrollo del modelo, desde la preparación del conjunto de datos hasta la evaluación. Además, el uso de recursos como A/B testing ha contribuido a demostrar la efectividad del modelo en condiciones del mundo real. Este enfoque metodológico ha permitido una gestión más eficiente y sistemática del ciclo de vida del modelo.

AGRADECIMIENTOS

Expresamos nuestros agradecimientos a la Universidad Nacional de Loja y a la Carrera de Ingeniería en Computación por el apoyo

brindado para la realización de este trabajo. El presente artículo forma parte de las actividades realizadas dentro de la materia de “Composición de Textos Científicos en Ingeniería” en la Carrera de “Computación”, en el periodo de “octubre 2023 – marzo 2024”.

DECLARACIÓN DE LA IA GENERATIVA Y TECNOLOGÍAS ASISTIDAS POR IA EN EL PROCESO DE REDACCIÓN:

En la elaboración de este artículo, se ha empleado tecnología de inteligencia artificial generativa (IAG) como herramienta de apoyo para mejorar y perfeccionar la redacción. Específicamente, se han utilizado ChatGPT, BARD y Perplexity. Es importante destacar que la IAG se ha utilizado únicamente como asistente, y que los autores han revisado y editado cuidadosamente el texto generado por estas herramientas.

Esta declaración tiene como objetivo ser transparente sobre el uso de la IAG en la redacción del artículo, y a la vez, enfatizar que la responsabilidad final del contenido y las ideas expresadas recae en los autores.

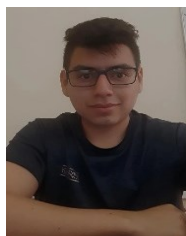
REFERENCIAS

- [1] L. Abarca-Guerrero, G. Maas, y W. Hogland, “Desafíos en la gestión de residuos sólidos para las ciudades de países en desarrollo”, *Revista Tecnología en Marcha*, vol. 28, núm. 2, p. ág. 141-168, jun. 2015, doi: 10.18845/TM.V28I2.2340.
- [2] S. Albawi, T. A. Mohammed, y S. Al-Zawi, “Understanding of a convolutional neural network”, *Proceedings of 2017 International Conference on Engineering and Technology, ICET 2017*, vol. 2018-January, pp. 1–6, mar. 2018, doi: 10.1109/ICENGTECHNOL.2017.8308186.
- [3] T.-B. Centeno *et al.*, “Herramientas de corte para optimizar parámetros de clasificación de especies maderables con redes neuronales convolucionales”, *Rev Biol Trop*, vol. 71, núm. 1, ene. 2023, doi: 10.15517/REV.BIOL.TROP.V71I1.51310.
- [4] J. R. Lapo Jima, “Detección de residuos sólidos reciclables utilizando redes neuronales convolucionales y PyTorch”, Bachelor Thesis, Universidad Nacional de Loja, Loja, 2024.
- [5] H. Abdu y M. H. Mohd Noor, “A Survey on Waste Detection and Classification Using Deep Learning”, *IEEE Access*, vol. 10, pp. 128151–128165, 2022, doi: 10.1109/ACCESS.2022.3226682.
- [6] J. Giraldo, “Clasificador de residuos sólidos haciendo uso de deep learning.”, Bachelor Thesis, Universidad Autónoma de Occidente, Santiago de Cali, Colombia, 2022, <https://hdl.handle.net/10614/14209>.
- [7] P. F. Proença y P. Simões, “TACO: Trash Annotations in Context for Litter Detection”, mar. 2020, doi:10.48550/arXiv.2003.06975.
- [8] V. Dewufl, “Application of machine learning to waste management: identification and classification of recyclables”, 2017, doi: 10.13140/RG.2.2.26805.50406.
- [9] J. O. Lim, “Comparison of Re-trained CNN Models from Pytorch, Keras, and Tensorflow Frameworks for Image Waste Classification”, *Theses and Dissertations (All)*, ene. 2019.
- [10] S. Karthikeyan, M. Sivakumar, A. Jeysiva, y C. Maheshkumar, “Application of Deep Learning for Solid Waste Trash Classification using Deep CNN”, pp. 126–131, 2021.
- [11] F. Núñez, “Dry Waste Segregation using Computer Vision”, *Proceedings of the 2022 39th IEEE Central America and Panama Student Convention Conference, CONESCAPAN*

- 2022, doi: 10.1109/CONESCAPAN56456.2022.9959238.
- [12] M. S. Saavedra Mayorga, "Implementación de un sistema de detección de residuos reciclables basado en visión computacional", Bachelor Thesis, Universidad de Chile, Santiago de Chile, Chile, 2020, <https://repositorio.uchile.cl/handle/2250/178007>.
- [13] M. Salina, "Deep Learning aplicado al procesamiento de imágenes para la detección de objetos reciclables.", Bachelor Thesis, Universidad Nacional Arturo Jauretche, Buenos Aires, Argentina, 2021, <https://biblioarchivo.unaj.edu.ar/mostrar/pdf/scvvsdf/erwe/fc8a4f46c475c1313b5355ff638ccb653637aed>.
- [14] G. E. Sakr, M. Mokbel, A. Darwich, M. N. Khneisser, y A. Hadi, "Comparing deep learning and support vector machines for autonomous waste sorting", *2016 IEEE International Multidisciplinary Conference on Engineering Technology, IMCET 2016*, pp. 207–212, dic. 2016, doi: 10.1109/IMCET.2016.7777453.
- [15] D. L. G. Vargas y A. K. T. Saavedra, "Clasificador de residuos sólidos para la i.e. Juan XXIII del municipio de Algeciras con aplicación de Machine Learning", *Revista Sennova: Revista del Sistema de Ciencia, Tecnología e Innovación*, dic. 2022, doi: 10.23850/23899573.5409.
- [16] M. Satvilkar, "Image Based Trash Classification using Machine Learning Algorithms for Recyclability Status", Master's Thesis, National College of Ireland, Dublin, 2018, <https://norma.ncirl.ie/3422/>.
- [17] R. Sultana, R. D. Adams, Y. Yan, P. M. Yanik, y M. L. Tanaka, "Trash and Recycled Material Identification using Convolutional Neural Networks (CNN)", *Conference Proceedings - IEEE SOUTHEASTCON*, vol. 2020-March, mar. 2020, doi: 10.1109/SOUTHEASTCON44009.2020.9249739.
- [18] J. M. Peña Lorenzo, "Aplicación de técnicas de aprendizaje profundo (deep learning) para la detección de objetos en Industria 4.0", Master's Thesis, Universidad de Valladolid, Valladolid, España, 2020, <http://uvadoc.uva.es/handle/10324/43255>.
- [19] J. F. Guerrero Citelly, "Traductor de lenguaje de señas portatil por medio de reconocimiento de imágenes", Bachelor Thesis, Universidad de los Andes, Colombia, 2018, <http://hdl.handle.net/1992/39076>.
- [20] Compri Michele, "Multi-label remote sensing image retrieval based on deep features", Bachelor Thesis, Università Degli Studi Di Trentp, Italia, 2016, <https://imatge.upc.edu/web/sites/default/files/pub/xCompri.pdf>.



Oscar M. Cumbicus-Pineda, is a Systems Engineer and Master in Intelligent Systems, he is currently a Professor of the Computer Science Program at the National University of Loja, and his area of interest is artificial intelligence.



Jhandry Lapo, is a Computer Science student at the Universidad Nacional de Loja. His research interests focus on topics related to intelligent systems and the development of artificial intelligence models.