



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CHILPANCINGO

Maestría en Ciencias de la Ingeniería

**ARQUITECTURA OPTIMIZADA DE REDES NEURONALES
RESIDUALES PARA LA CLASIFICACIÓN DE RESIDUOS
SÓLIDOS URBANOS CON PROCESAMIENTO DE IMÁGENES**

TESIS

**PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA INGENIERÍA**

Presenta

Víctor Manuel Romero Juárez

Directora

Dra. Mirna Castro Bello
INSTITUTO TECNOLÓGICO DE CHILPANCINGO

Codirector

Dr. Jorge Alberto Fuentes Pacheco
CENTRO NACIONAL DE INVESTIGACIÓN Y
DESARROLLO TECNOLÓGICO

Revisor 1

Dr. Carlos Virgilio Marmolejo Vega
INSTITUTO TECNOLÓGICO DE CHILPANCINGO

Revisor 2

Mtro. Sergio Ricardo Zagal Barrera
INSTITUTO TECNOLÓGICO DE CHILPANCINGO

Chilpancingo De los Bravo, Guerrero, mayo 2025

Agradecimientos

A dios por la sabiduría y fuerza para completar cada una de las tareas.

A mi mamá una de las personas que más amo, quien me ha apoyado de inicio a fin en esta trayectoria y me ha enseñado a nunca rendirme y siempre dar lo mejor de mí dónde quiera que me encuentre.

A mi papá una de las personas que más amo, quien me ha enseñado el valor del trabajo y disciplina y me ha apoyado en esta etapa.

A mi hermana, quien con sus locuras me hizo sonreír en momentos complicados.

A la Dra. Mirna Castro Bello con todo mi cariño, por la confianza, apoyo y atención que me ha brindado, quien como mi directora de proyecto me ha guiado y ha sido un gran ejemplo para seguir y con quien he tenido maravillosas experiencias en esta trayectoria.

Al Dr. Jorge Alberto Fuentes Pacheco con todo mi cariño, por acompañarme y aconsejarme durante el desarrollo del proyecto, con quien aprendí el cómo superar los miedos.

Al Dr. Carlos Virgilio Marmolejo Vega con todo mi cariño, por su confianza en que lograría este objetivo y que soy capaz de lograr grandes cosas.

Al Dr. Cornelio Morales Morales con todo mi cariño, quien con sus consejos y sabiduría me enseñó que soy capaz de alcanzar grandes cosas.

A los directivos del Instituto Tecnológico de Chilpancingo, la M.D.I.S. Areli Bárcenas Nava y el Mtro. Sergio Ricardo Zagal Barrera por su disposición en brindar apoyo cuando se requiriera.

A la Maestría en Ciencias de la Ingeniería, donde aprendí muchas cosas y amplió mi visión en este mundo de la investigación.

A la SECIHTI, quien me otorgó la beca para poder alcanzar esta meta.

A Liz y Dominic por su grandiosa amistad y por el apoyo siempre brindado.

A mis profesores y compañeros quienes me acompañaron durante mi formación.



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Chilpancingo
División de Estudios de Posgrado e Investigación

Chilpancingo, Guerrero **12/mayo/2025**
Oficio No. 328/2025

Víctor Manuel Romero Juárez
Candidato a Maestro en Ciencias de la Ingeniería
PRESENTE

Por este conducto me es grato comunicarle que el Comité Tutorial asignado a su trabajo de tesis titulado **"Arquitectura optimizada de redes neuronales residuales para la clasificación de residuos sólidos urbanos"**, ha informado a esta División de estudios de Posgrado e Investigación que están de acuerdo con el trabajo presentado. Por lo anterior, se le autoriza a que proceda con la impresión definitiva de su trabajo de tesis.

ATENTAMENTE
Excelencia en Educación Tecnológica®

MIRNA CASTRO BELLO
JEFA DE LA DIVISIÓN DE ESTUDIOS DE
POSGRADO E INVESTIGACIÓN



C.c.p. Subdirección Académica
C.c.p. Archivo
MCB*rlgr



2025
Año de
La Mujer
Indígena

Av. José Francisco Ruiz Massieu No. 5, Colonia Villa Moderna, Chilpancingo
de los Bravo, Guerrero. México. Tel. (747) 45 4 1300, Ext. 1310
Email: depi@chilpancingo.tecnm.mx <http://chilpancingo.tecnm.mx>
<https://www.facebook.com/TecNMcampusChilpancingo>



DRA. MIRNA CASTRO BELLO

JEFA DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO

E INVESTIGACIÓN DEL ITCHILPANCINGO

PRESENTE

Por este medio los integrantes del comité Tutorial del **C. Víctor Manuel Romero Juárez** con número de control **G16520276** de la Maestría en Ciencias de la Ingeniería, le informamos que hemos revisado el trabajo de tesis de grado denominado **“Arquitectura optimizada de redes neuronales residuales para la clasificación de residuos sólidos urbanos”**, una vez que se han atendido todas las observaciones que se le indicaron, hemos acordado aceptar el documento por lo que solicitamos la autorización de impresión definitiva

Sin más por el momento, le enviamos un cordial saludo.

Directora



Dra. Mirna Castro Bello

Codirector



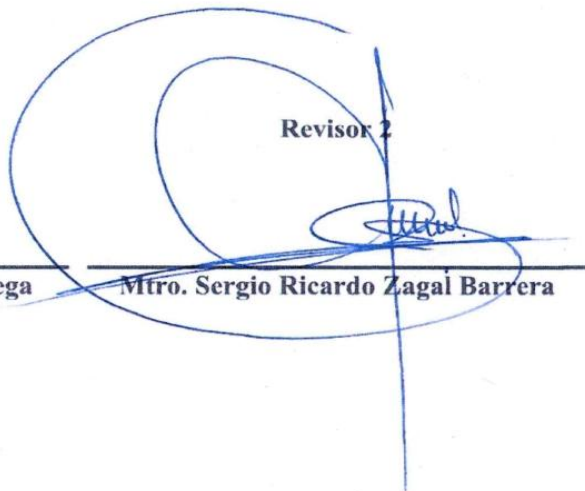
Dr. Jorge Alberto Fuentes Pacheco

Revisor 1



Dr. Carlos Virgilio Marmolejo Vega

Revisor 2



Mtro. Sergio Ricardo Zagai Barrera

Índice de cuadros	VII
Índice de figuras.....	VIII
Resumen.....	IX
Abstract.....	X
Capítulo I. Introducción	1
Capítulo II. Antecedentes.....	4
2.1 En el mundo	5
2.2 En América.....	6
2.3 En México	7
2.4 Conceptos.....	7
Capítulo III. Justificación.....	10
Capítulo IV. Objetivos	12
4.1 General.....	12
4.2 Específicos	12
Capítulo V. Metodología	13
5.1 Materiales.....	13
5.2. Metodología	13
5.2.1 Creación del <i>dataset</i>	15
5.2.1.1 Recolección de imágenes	15
5.2.1.2 Eliminación de duplicados	18
5.2.1.3 Creación del dataset orgánicos e inorgánicos	18
5.2.2 Análisis de algoritmos CNN para la clasificación de RSU.....	20
5.2.2.1 Revisión sistemática de la literatura	20
5.2.3 Preparación y generación de datos para el entrenamiento del modelo	21
5.2.3.1 Conversión de imágenes al formato RGB	21

5.2.3.2	División del conjunto de datos en subconjuntos.....	21
5.2.3.3	Aumento de imágenes en línea	21
5.2.4	Entrenamiento e identificación de redes residuales con la mayor extracción de características	22
5.2.4.1	Desarrollo de arquitectura basada en ResNet18	22
5.2.5	Experimentación y ajuste de la reducción de parámetros del modelo residual propuesto23	
5.2.5.1	Estructura de la red residual propuesta	23
5.2.6	Evaluación del algoritmo de red residual.....	31
5.2.6.1	Análisis de parámetros.....	31
Capítulo VI.	Resultados	33
6.1	<i>Dataset</i> de RSU.....	33
6.2	Algoritmos CNN para la clasificación de RSU	35
6.3	Preparación y generación de datos para el entrenamiento del modelo CNN.....	37
6.4	Entrenamiento e identificación de redes residuales	38
6.5	Experimentación y ajustes de parámetros	39
6.6	Evaluación del algoritmo de red residual.....	41
Capítulo VII.	Discusión	45
Capítulo VIII.	Conclusiones	50
Referencias bibliográficas.....		52
IX. Anexos		58
Anexo 1.	Código para la detección de imágenes duplicadas	58
Anexo 2.	Script de conversión de formato de color de imágenes a RGB de RSU inorgánicos	59
Anexo 3.	Script de conversión de formato de color de imágenes a RGB de RSU orgánicos...	59
Anexo 4.	División del conjunto de datos en entrenamiento y validación.....	59
Anexo 5.	División del <i>dataset</i> en entrenamiento y validación	60

Anexo 6. Configuración del aumento de datos en línea	60
Anexo 7. Código de R3sNet	61
Anexo 8. Productos académicos	62

Índice de cuadros

<i>Tabla 1 Descripción de las imágenes obtenidas de cada Dataset consultado</i>	<i>16</i>
<i>Tabla 2 Valores usados para el aumento de datos en fase de entrenamiento</i>	<i>22</i>
<i>Tabla 3 Análisis de las diferentes arquitecturas aplicadas en la clasificación de RSU.</i>	<i>35</i>
<i>Tabla 4 Hiperparámetros utilizados en R3sNet.</i>	<i>39</i>
<i>Tabla 5 Comparación de parámetros y GFLOPS entre R3sNet y ResNet50.</i>	<i>40</i>
<i>Tabla 6 Métricas de rendimiento para R3sNet y ResNet50</i>	<i>41</i>

Índice de figuras

<i>Figura 1</i>	<i>Proceso metodológico.....</i>	<i>14</i>
<i>Figura 2</i>	<i>Muestra de las imágenes de los seis datasets utilizados, a: TrashNet, b: Garbage Huawei, c: Garbage classification data, d: Garbage dataset, e: Garbage Classification y f: Garbage Pictures for Classification.</i>	<i>19</i>
<i>Figura 3</i>	<i>Muestras del conjunto de datos para evaluar la red propuesta: a: residuos orgánicos y b: residuos inorgánicos.</i>	<i>20</i>
<i>Figura 4</i>	<i>Dataset RSU orgánicos.</i>	<i>33</i>
<i>Figura 5</i>	<i>Dataset RSU inorgánicos.</i>	<i>34</i>
<i>Figura 6</i>	<i>Dataset de prueba orgánicos.</i>	<i>37</i>
<i>Figura 7</i>	<i>Dataset de prueba inorgánicos.</i>	<i>37</i>
<i>Figura 8</i>	<i>Diagrama de arquitectura R3sNet.....</i>	<i>38</i>
<i>Figura 9</i>	<i>Cantidad de parámetros de R3sNet.</i>	<i>40</i>
<i>Figura 10</i>	<i>Gráfico que muestra la media y la desviación estándar de R3sNet en 10 ejecuciones.</i>	<i>42</i>
<i>Figura 11</i>	<i>Gráfico de la media y la desviación estándar utilizando ResNet50 preentrenado en 10. ejecuciones.</i>	<i>43</i>
<i>Figura 12</i>	<i>Matriz de confusión de los modelos (a) R3sNet y (b) ResNet50 preentrenado.</i>	<i>44</i>

Resumen

La acumulación de Residuos Sólidos Urbanos (RSU) representa un desafío crítico para la sociedad y el gobierno, y su separación es esencial para garantizar una utilización sostenible. Sin embargo, persiste el desafío de la clasificación manual, una labor que, lamentablemente, se mantiene como una práctica inherentemente lenta y con considerables ineficiencias. Frente a esta realidad, los progresos exponenciales en el campo de la inteligencia artificial, y de manera destacada el aprendizaje automático, emergen como soluciones alternativas que no solo prometen mayor precisión, sino también una eficiencia superior para optimizar radicalmente este flujo de trabajo. En este marco, la contribución que aquí se presenta detalla la gestación de una red neuronal profunda, enriquecida con módulos residuales, diseñada y entrenada de extremo a extremo. Este modelo está específicamente orientado a la clasificación binaria de los RSU y, crucialmente, exhibe la habilidad de ejecutar inferencias a una velocidad significativamente incrementada. Mediante una metodología de cinco fases: 1. Creación del conjunto de datos (dataset) de RSU: recolección de imágenes, eliminación de duplicados, creación del conjunto de datos orgánicos e inorgánicos; 2. Analizar algoritmos de procesamiento de imágenes; 3. Preparación y generación de datos: Conversión de imágenes a formato RGB, división del conjunto de datos de entrenamiento en subconjuntos y aumento de imágenes en línea; 4. Entrenar e identificar las redes neuronales residuales para la clasificación de RSU orgánicos e inorgánicos con la mayor extracción de características: Desarrollo de arquitectura basada en ResNet18; 5. Experimentar y ajustar la reducción de parámetros del modelo residual propuesto: Modificar la estructura de la red residual y ajustar hiperparámetros claves; y 6. Evaluar el algoritmo de clasificación: Precisión, matriz de confusión, f1-score, recall y exactitud. Se obtuvo un 87% para la clasificación de residuos orgánicos y un 94% para los inorgánicos. R3sNet destaca frente al modelo pre-entrenado ResNet50 en la clasificación de ambas categorías (orgánicos e inorgánicos) con una ventaja de hasta un 6%, y una disminución en la carga computacional y complejidad del modelo. En concreto, se hizo una reducción del 98.60% en el número de hiperparámetros y un recorte del 65.17% en los GFLOPS en comparación directa con ResNet50.

Palabras clave: CNN, clasificación de residuos, arquitectura optimizada, residual networks, deep learning

Abstract

The accumulation of municipal solid waste (MSW) represents a critical challenge for society and government, and its separation is essential to ensure its sustainable use. However, the challenge of manual sorting persists, a task that, unfortunately, remains inherently slow and fraught with considerable inefficiencies. Faced with this reality, exponential advances in the field of artificial intelligence, and particularly machine learning, are emerging as alternative solutions that promise not only greater accuracy but also greater efficiency to radically optimize this workflow. Within this framework, the contribution presented here details the development of a deep neural network, enriched with residual modules, designed and trained end-to-end. This model is specifically oriented to the binary classification of MSW and, crucially, exhibits the ability to perform inferences at a significantly increased speed. Using a five-phase methodology: 1. Creation of the MSW dataset: image collection, duplicate removal, creation of the organic and inorganic dataset; 2. Analysis of image processing algorithms; 3. Data preparation and generation: Conversion of images to RGB format, division of the training dataset into subsets, and online image augmentation; 4. Application of convolutional neural networks for the classification of organic and inorganic MSW: development of a ResNet18 based architecture; and 5. Evaluation of the classification algorithm: Precision, confusion matrix, f1-score, recall, and accuracy. A score of 87% was obtained for the classification of organic waste and 94% for inorganic waste. R3sNet stands out compared to the pre-trained ResNet50 model in the classification of both categories (organic and inorganic), with an advantage of up to 6%, and a decrease in the computational load and complexity of the model. Specifically, there was a 98.60% reduction in the number of hyperparameters and a 65.17% reduction in GFLOPS compared directly to ResNet50.

Keywords: CNN, waste classification, optimized architecture, residual networks, deep learning

Capítulo I. Introducción

El rápido aumento de la industrialización y la urbanización ha producido grandes volúmenes de RSU destacando la importancia de perfeccionar la categorización. Este suceso evidencia el incremento de la actividad humana y los retos vinculados a la administración de residuos. En 2020 se generaron aproximadamente 2,010 millones de toneladas de RSU, según el Banco Mundial. Estas cifras son significativas, y las previsiones indican que para 2030, esta cantidad aumentará a 2,590 millones y 3,400 millones de toneladas para el 2050, de los cuales, alrededor del 40% se vierten en vertederos, el 19% se recicla y el 11% se procesa a través de incineración (Kaza et al., 2018; Lin et al., 2022; Xia et al., 2022; Zhang et al., 2021). Esto muestra la importancia de optimizar los procesos, en particular los vinculados al reciclaje y la disminución del uso de vertederos. Es esencial una clasificación eficaz para fomentar la sostenibilidad del medio ambiente y potenciar el bienestar social. Mejorando los procesos de reciclaje se puede reducir la huella ecológica, minimizar la dependencia de los vertederos y fomentar la economía circular, que busca reutilizar recursos y reducir los residuos. Esta perspectiva concuerda con estrategias mundiales como los Objetivos de Desarrollo Sostenible (ODS), en particular los ODS 11, 12 y 13, teniendo como objetivo enfrentar retos como la urbanización desmedida, el consumo desmedido y el cambio climático.

La sustitución de la clasificación manual de residuos por sistemas automatizados basados en IA se revela imprescindible para reducir los altísimos costos laborales inherentes al reciclaje tradicional (Chhabra et al., 2024; Sayem et al., 2025). Además, los algoritmos avanzados de aprendizaje automático potencian la precisión del proceso de separación de desechos al abordar complejos retos de reconocimiento y clasificación mediante visión por computador (Chhabra et al., 2024; Xia et al., 2022b). En este contexto, las Redes Neuronales Convolucionales (CNN) con estructuras

profundas concebidas para aprender jerarquías espaciales de características visuales mediante capas convolucionales sucesivas y una capa totalmente conectada final, se consolidan como la metodología dominante en clasificación de imágenes. Las arquitecturas modernas de CNN (por ejemplo, VGG, ResNet, DenseNet o EfficientNet) añaden profundidades y refinamientos que mejoran la extracción de rasgos y optimizan la precisión final; de hecho, estudios recientes han reportado modelos optimizados (DenseNet121 modificado) con precisiones superiores al 99% en conjuntos estándar de imágenes de residuos (Xia et al., 2022b; Zhao et al., 2024).

Varios estudios documentan la aplicación de redes neuronales profundas en la clasificación de residuos sólidos municipales, logrando resultados notables. Por ejemplo, Mao et al. (2021) diseñaron una CNN optimizada basada en DenseNet121, en la cual se empleó un algoritmo genético para afinar la capa totalmente conectada, alcanzando una precisión del 99.6% (red de 8 millones de parámetros). Zhang et al. (2021) propusieron un modelo de clasificación de imágenes de basura basado en DenseNet169 con estrategia de transfer learning, construyendo el conjunto NWNUTRASH (fondos diversos y distribución equilibrada); emplearon un 70% de imágenes para entrenamiento y 30% para prueba, logrando más del 82% de precisión. Otros estudios recientes reportan precisiones comparables: por ejemplo, Shi et al. (2021) alcanzaron un 92.6% de exactitud con una CNN híbrida multicapa en el conjunto TrashNet, mientras que Chhabra et al. (2024) obtienen 93.3% al separar desechos orgánicos vs. reciclables mediante un DCNN mejorado tras transfer learning.

La clasificación binaria de los RSU presenta ventajas en términos de eficiencia del reciclaje, reducción de costos y fomento de la economía circular. Dicha estrategia resulta especialmente eficaz en entornos con recursos limitados, escasos niveles de conciencia ambiental o un elevado volumen de residuos. Por el contrario, en contextos donde la generación de residuos es reducida,

la infraestructura es limitada o los objetivos de gestión son esencialmente básicos, no resulta necesaria la implementación de sistemas de clasificación multiclase. En tales casos, la clasificación binaria puede construir una solución eficiente y efectiva, este tipo de clasificación se erige como una estrategia robusta y efectiva, sentando una base fundamental en los sistemas reales de gestión de RSU que, posteriormente, podría expandirse mediante esquemas jerárquicos más detallados.

Esta investigación tuvo como objetivo el desarrollo de una red neuronal profunda y ligera con módulos residuales entrenados integralmente para la clasificación binaria de RSU, con capacidad para una inferencia más rápida.

Capítulo II. Antecedentes

En el marco de la sustentabilidad ambiental y de los procesos de urbanización, privatización y descentralización, la gestión de los RSU constituye hoy una preocupación de singular importancia por su impactos directos e indirectos, algunos de ellos irreversibles y permanentes en el medio ambiente (aire, agua, tierra, paisaje) y en la salud. El deterioro de la calidad de las aguas superficiales por escurrimiento de los lixiviados, que resulta del contenido líquido de los residuos más el arrastre de aguas de lluvias, y por la acción de los líquidos percolados en las napas freáticas. Así como la generación de emisiones globales de gases de efecto invernadero que representan el 3% equivalente a 1,32 millones de toneladas de CO₂ (Rondon Toro et al., 2016). La acumulación de basura en vertederos y sitios no controlados provoca contaminación de suelos y aguas, además de emisiones de gases de efecto invernadero que amenazan la salud humana y la biodiversidad (Kaza et al., 2018). En suma, el panorama global evidencia una crisis creciente: los sistemas de gestión actuales son incapaces de seguir el ritmo de generación, lo que compromete los objetivos de desarrollo sostenible y la mitigación de la contaminación.

En este contexto, la inteligencia artificial (IA) ha emergido como estrategia prometedora para optimizar el manejo de residuos. En particular, los sistemas de aprendizaje profundo basados en CNN que destacan por su capacidad de procesar imágenes de basura y clasificarlas automáticamente. La literatura reciente subraya que las CNN pueden lograr rendimientos muy altos en la detección y categorización de RSU (Bhattacharya et al., 2024).

La situación latinoamericana refleja similares tendencias y desafíos, en 2021 América Latina y el Caribe generó alrededor de 230,5 millones de toneladas de RSU, Brasil (73.3 millones t) y México (47.7 millones t) son los países con mayor contribución a ese total (Statista, 2024).

En México, las cifras oficiales confirman la urgencia de innovar en la gestión de los residuos. La Secretaría de Medio Ambiente estimó una generación per cápita de 0.944 kg/hab por día, equivalente a un total de 120,128 toneladas diarias de RSU (SEMARNAT, 2020). Gran parte se maneja actualmente con métodos tradicionales (vertederos, rellenos sanitarios) y con bajos índices de separación, lo que acentúa la presión ambiental. La literatura nacional reconoce estos retos y apunta hacia la IA como vía de mejora. Yáñez y Benavides (2024) subrayan que dispositivos inteligentes, combinados con redes neuronales, pueden “separar diferentes materiales de manera precisa” y aumentar drásticamente la eficiencia del reciclaje. Además, la automatización con IA promete alivianar la labor humana en plantas de tratamiento, reduciendo la carga operativa y el costo laboral. Diversas investigaciones reportan propuestas para la clasificación de RSU utilizando IA, particularmente CNN.

2.1 En el mundo

En India, Malik et al. (2022) desarrollaron una red neuronal convolucional multicapa híbrida con pocos parámetros utilizando imágenes de 64×64 píxeles obtenidas de TrashNet, logrando una precisión del 92,6%. En China, Zhang et al. (2021) proponen un modelo de clasificación de imágenes de residuos reciclables que aplica técnicas de aumento de imágenes basadas en una arquitectura de 18 capas (ResNet18) con una precisión del 95.87 %. Shi et al. (2021) propusieron una red neuronal convolucional inspirada en VGGNet para la clasificación de residuos. La MLH-CNN se optimizó con menos parámetros y módulos híbridos, y se utilizó TrashNet como conjunto de datos, logrando una precisión del 92.6 %. También Lin et al. (2022) desarrollaron un modelo de clasificación de residuos reciclables mediante aprendizaje profundo llamado RWNet basado en varias estructuras ResNet, utilizaron el conjunto de datos TrashNet con técnicas de aumento de imágenes, logrando una precisión del 88%; y Yang et al. (2022) propusieron una pequeña CNN

para la clasificación de residuos, con un algoritmo adaptativo de imagen para equilibrar el brillo del fondo durante la fase de preprocesamiento. También implementaron un método de reemplazo de umbral diseñado para mitigar el ruido generado por las sombras. Utilizaron el operador Canny para recortar el fondo blanco y reducir la interferencia visual, logrando una precisión del 96.77 % en su conjunto de datos personalizado y del 93.72 % en el conjunto de datos TrashNet.

En Bangladesh, Islam et al. (2025) desarrollaron un modelo de Red Neuronal Profunda de Clasificación Ecológica (ECCDN-Net) para la clasificación binaria (orgánica e inorgánica) de imágenes de residuos utilizando componentes de Densenet201 y Resnet18. Utilizaron 24,705 imágenes, logrando una precisión de clasificación del 96.10 %. Azis et al. (2020) elaboran un modelo CNN para la clasificación de residuos. En concreto, el modelo Inception-v3 se entrenó con un conjunto de datos obtenidos de diferentes fuentes, logrando una precisión de clasificación del 92.5 % y Mao et al. (2021) diseñaron una CNN optimizada basada en DenseNet121, en la cual se empleó un algoritmo genético para afinar la capa totalmente conectada, alcanzando una precisión del 99,6% (red de 8 millones de parámetros).

2.2 En América

En Estados Unidos, Roy y Marellapudi (2023) desarrollaron un modelo de red neuronal utilizando el optimizador Adam alcanzando un 84% de precisión. Lapo y Cumbicus-Pineda (2024) propusieron un modelo CNN utilizando PyTorch y el conjunto de datos Taco Trash Dataset, obtuvieron una precisión de 97%.

2.3 En México

Becíez Salazar (2020) desarrolló una aplicación para la identificación de botellas PET, y los materiales inorgánicos como el papel y cartón, en el que se hace uso de una red neuronal convolutiva.

2.4 Conceptos

Residuos: La Ley General del Equilibrio Ecológico y la Protección al Ambiente publicada en el Diario Oficial de la Federación el día 01 de abril del 2024, artículo 3, Fracción XXXII, define como residuo a “Cualquier material generado en los procesos de extracción, beneficio, transformación, producción, consumo, utilización, control o tratamiento cuya calidad no permita usarlo nuevamente en el proceso que lo generó” (LGEEPA, 2024).

Residuos Sólidos Urbanos (RSU): La Ley para la Prevención y Gestión Integral de los Residuos define los define como los generados en las casas habitación, que resultan de la eliminación de los materiales que utilizan en sus actividades domésticas, de los productos que consumen y de sus envases, embalajes o empaques los residuos que provienen de cualquier otra actividad dentro de establecimientos o en la vía pública que genere residuos con características domiciliarias, y los resultantes de la limpieza de las vías y lugares públicos (LGPGIR, 2023).

Inteligencia Artificial (IA): Es el campo de la informática que se centra en crear sistemas (agentes) capaces de percibir su entorno, razonar sobre él y realizar acciones para alcanzar objetivos, imitando aspectos de la inteligencia humana (Russell & Norvig, 2020).

Red Neuronal: Modelo computacional inspirado en las redes neuronales biológicas. Consiste en capas de unidades sencillas (neuronas) interconectadas mediante enlaces ponderados. La red

aprende a realizar tareas ajustando estos pesos, de modo que puede aproximar funciones complejas que mapean entradas a salidas (Han et al., 2018).

Deep Learning: Subdisciplina del aprendizaje automático que emplea redes neuronales con múltiples capas (“profundas”) para modelar patrones complejos en los datos. Estas redes aprenden representaciones jerárquicas: las capas inferiores capturan características simples y las capas superiores combinan estas características en patrones cada vez más abstractos (Goodfellow et al., 2016).

Redes Residuales: Arquitecturas que incluyen conexiones de atajo (o conexiones residuales) que omiten una o más capas. Conexiones que suman la entrada de un bloque directamente a su salida, lo cual ayuda a estabilizar el entrenamiento y a entrenar eficientemente redes muy profundas (He et al., 2015).

Redes Neuronales Convolucionales o convnets (CNN): Arquitectura de red neuronal especialmente adaptada a la estructura de las señales visuales. La idea clave de las CNN es fragmentar la imagen de entrada en pequeños fragmentos y luego procesar cada fragmento de forma independiente e idéntica (Torralba et al., 2024).

ResNet: Arquitectura específica de red neuronal profunda introducida para reconocimiento de imágenes. Se basa en la idea de bloques residuales (con conexiones de atajo) y demostró permitir el entrenamiento efectivo de redes muy profundas (más de 100 capas) con alto desempeño en tareas de visión (He et al., 2015).

Tensorflow: Plataforma de aprendizaje automático de código abierto, gratuita y basada en Python, desarrollada principalmente por Google. Al igual que NumPy, el propósito principal de

TensorFlow es permitir que los ingenieros e investigadores manipulen expresiones matemáticas sobre tensores numéricos (Chollet, 2021).

Keras: API de aprendizaje profundo para Python, construida sobre TensorFlow, que proporciona una forma conveniente de definir y entrenar cualquier tipo de modelo de aprendizaje profundo. Keras se desarrolló inicialmente para la investigación, con el objetivo de permitir una rápida experimentación de aprendizaje profundo (Chollet, 2021).

Capítulo III. Justificación

Las actividades humanas están causando una cantidad cada vez mayor de contaminación, hasta el punto de ser, hoy en día, reconocidas como el mayor peligro para la salud. El cambio climático amplificará riesgos existentes y creará nuevos riesgos para los sistemas naturales y humanos, los cuales están distribuidos de manera desigual y son generalmente mayores en los países en desarrollo y para las personas y comunidades desfavorecidas (UNEP, 2019).

El 25 de septiembre de 2015, los líderes mundiales adoptaron un conjunto de objetivos globales para erradicar la pobreza, proteger el planeta y asegurar la prosperidad para todos como parte de una nueva agenda de desarrollo sostenible. En este contexto, el objetivo 11: Ciudades y Comunidades Sostenibles, meta 11.6: de aquí a 2030, es reducir el impacto ambiental negativo per cápita de las ciudades, incluso prestando especial atención a la calidad del aire y la gestión de los desechos municipales y de otro tipo (ONU-HABITAT, 2018).

Todas las ciudades, independientemente de su tamaño y capacidad financiera, pueden mejorar su situación actual de gestión de residuos sólidos para convertirse en ciudades con manejo responsable de sus desechos, reducir el costo operacional y, al mismo tiempo, minimizar los impactos negativos sobre la salud y el medioambiente (ONU-HABITAT, 2018).

La clasificación binaria de RSU ofrece ventajas en términos de eficiencia en el reciclaje, la reducción de costos y el fomento a una economía circular, es funcional en escenarios que cuentan con recursos limitados, cuentan con bajos niveles de educación ambiental o generan altos volúmenes de residuos orgánicos. Por otro lado, donde la generación de residuos es baja, se cuenta con infraestructura limitada o los objetivos de gestión son prioritariamente básicos no es necesario

implementar sistemas de clasificación multiclase debido a que la clasificación binaria puede ser una solución eficiente y efectiva.

Con el propósito de eficientizar la caracterización y clasificación de los RSU, esta investigación propone el desarrollo de una arquitectura optimizada con redes residuales entrenada de inicio a fin para la clasificación binaria de RSU con capacidad de hacer inferencia más rápida y pueda aplicarse a otras áreas del conocimiento; por ejemplo, medir el nivel de madurez de frutos, en el área de la salud, medio ambiente y la biología.

Capítulo IV. Objetivos

4.1 General

- Desarrollar una arquitectura optimizada con redes residuales para la clasificación de residuos sólidos urbanos.

4.2 Específicos

1. Crear el *dataset* de RSU.
2. Analizar algoritmos de CNN para la clasificación de RSU.
3. Preparar y generar datos para el entrenamiento del modelo CNN.
4. Entrenar e identificar las redes neuronales residuales para la clasificación de RSU orgánicos e inorgánicos con la mayor extracción de características.
5. Experimentar y ajustar la reducción de parámetros del modelo residual propuesto.
6. Evaluar el algoritmo de la red residual.

Capítulo V. Metodología

5.1 Materiales

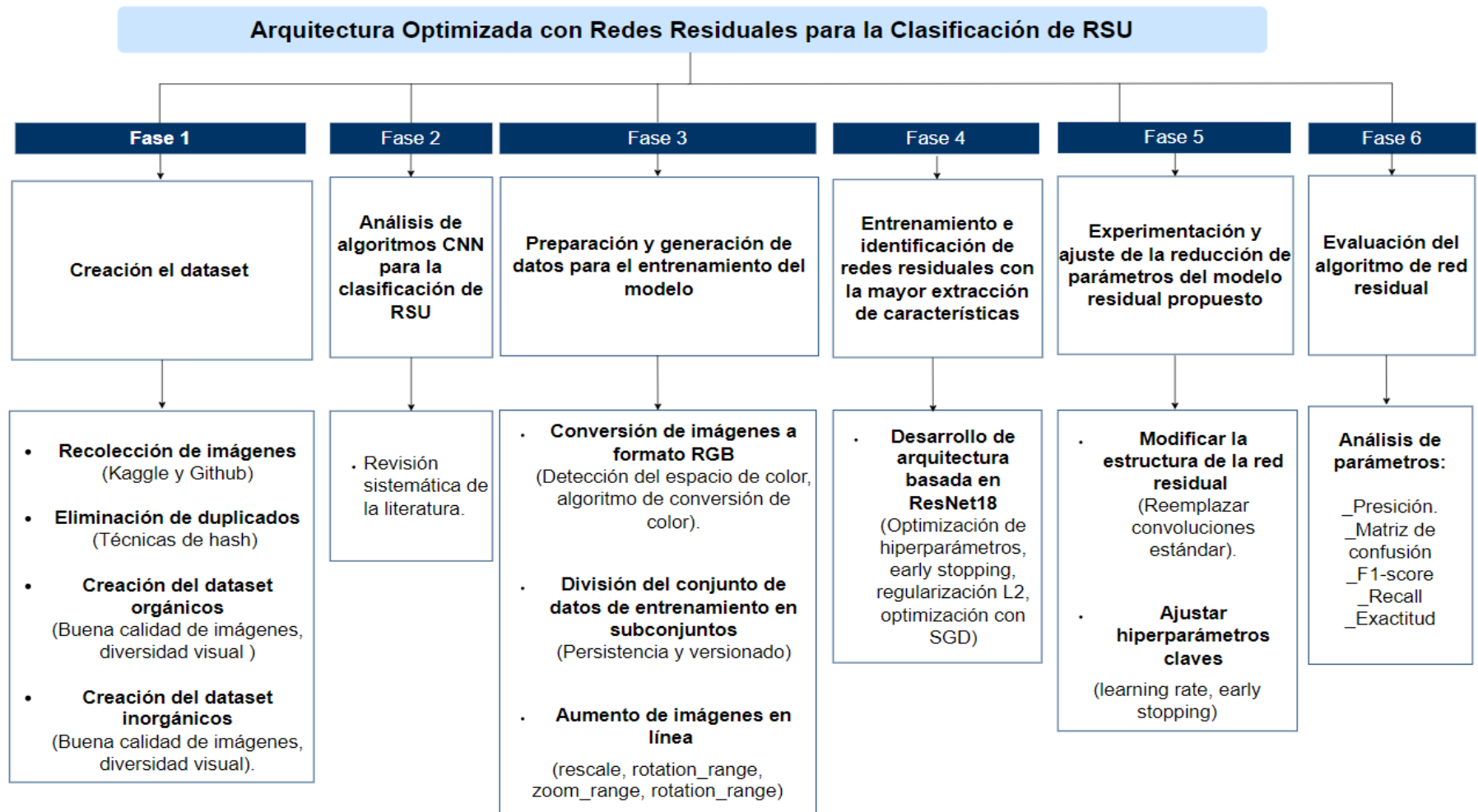
Para el desarrollo del modelo de red neuronal denominada R3sNet se utilizaron: Python versión 3.10.14, bibliotecas y entornos Tensorflow 2.10.1, Keras 2.10.0, Numpy 1.26.4, Pandas 1.4.4, Matplotlib 3.9.0, Seaborn 0.13.2, equipo de cómputo con procesador Intel Core i7, GPU NVIDIA GTX 1060Ti con 6 GB de memoria dedicada, memoria RAM de 23 GB y almacenamiento de 1 Tb de disco SSD.

5.2. Metodología

Para el desarrollo de la arquitectura optimizada con redes residuales para la clasificación de RSU, se aplicó el proceso metodológico, figura 1.

Figura 1

Proceso metodológico



5.2.1 Creación del *dataset*

5.2.1.1 Recolección de imágenes

Se consultaron múltiples bases de datos públicas para la composición del conjunto de datos de residuos, como *TrashNet* que considera residuos de plástico, metal, vidrio, papel y cartón (Thung, 2017), *Garbage_Huawei* que contiene residuos orgánicos, peligrosos e inorgánicos (Xiaohoua, 2023), *Waste Classification* data con residuos orgánicos e inorgánicos (Sekar, 2018), *Garbage Dataset* contiene imágenes de metal, vidrio, biológico, papel, baterías, basura, cartón, zapatos, ropa y plástico (Kunwar, 2024), *Garbage Classification* (Mohamed, 2020) con ilustraciones de batería, cartón, biológico, ropa, vidrio verde, metal, papel, plástico, zapatos, basura y vidrio blanco y *Garbage Pictures for Classification* que cuenta con las categorías de cartón, vidrio, metal, papel, plástico y basura (<https://www.kaggle.com/datasets/mascot9183/garbage-pictures-for-classification>, 2022), tabla 1. Las imágenes de este conjunto de datos contienen los residuos sobre un fondo controlado, lo que permite que la red neuronal se centre únicamente en las características inherentes de los residuos, garantizando que no haya sesgos causados por los fondos complejos sobre los que se colocan. Un sistema de clasificación de residuos entrenado con imágenes sobre fondos controlados es particularmente beneficioso en cintas transportadoras con iluminación constante y fondos uniformes, así como en máquinas de reciclaje donde los usuarios colocan manualmente sus residuos.

Tabla 1

Descripción de las imágenes obtenidas de cada Dataset consultado

Nombre dataset	Enlace	Cantidad imágenes
TrashNet	https://github.com/garythung/trashnet	993
Garbage_HuaWei	https://www.kaggle.com/datasets/xiaohoua/garbage-huawei	548
Waste Classification data	https://www.kaggle.com/datasets/techsaish/waste-classification-data	731
Garbage Dataset	https://www.kaggle.com/datasets/sumn2u/garbage-classification-v2	175
Garbage Classification (12 classes)	https://www.kaggle.com/datasets/mostafaabla/garbage-classification	275
Garbage Pictures for Classification	https://www.kaggle.com/datasets/mascoth9183/garbage-pictures-for-classification?select=Garbage+classification	339

Se consideraron los repositorios públicos de *Kaggle* por su variedad y calidad de imágenes en su *dataset* y *Github* debido a que en diferentes estudios aplicados a modelos de DL utilizaron *TrashNet* por la practicidad en la aplicación a un mayor número de datos.

Además, para la construcción del conjunto de datos utilizado en este estudio, se definieron criterios específicos que permitieron seleccionar y combinar diversos *dataset* existentes, asegurando la calidad y relevancia del *dataset* final. Los criterios considerados fueron:

- **Diversidad visual**

Se consideró por una alta variedad visual en los conjuntos de datos, incluyendo diferentes condiciones de iluminación, perspectivas, fondos y resoluciones con el propósito de mejorar la capacidad de generalización de los modelos.

- **Pertinencia temática**

Para garantizar la relevancia y aplicabilidad de cada imagen seleccionada, se incluyeron únicamente *datasets* cuyo contenido correspondía directamente a las clases orgánicas e inorgánicas.

- **Origen verificable y calidad**

Se priorizaron aquellos provenientes de fuentes confiables y reconocidas, tales como repositorios académicos, bases de datos oficiales, con la finalidad de asegurar la calidad y autenticidad del material visual.

- **Licencias y permisos de uso**

Se verificó cuidadosamente que todos los seleccionados contaran con licencias abiertas o permisos explícitos de uso que permitiesen su reutilización, modificación y distribución en este proyecto.

- **Balance de clases**

Se consideró la inclusión de *dataset* que ofrecieran una distribución equilibrada de imágenes entre las diferentes categorías o clases, minimizando sesgos y mejorando el desempeño predictivo del modelo.

- **Calidad y resolución de imágenes**

Se optó por *dataset* compuestos por imágenes de resolución y calidad adecuadas, garantizando que las características extraídas fueran óptimas para la tarea propuesta.

- **Tamaño del *dataset***

Se escogieron los que cuentan con un volumen significativo de imágenes, para que, tras la aplicación del proceso de filtrado, el conjunto final resultara suficientemente robusto para el entrenamiento efectivo de los modelos.

Estos criterios permitieron obtener un *dataset* final coherente, relevante y libre de redundancias visuales, optimizando así los resultados esperados en los modelos de DL.

5.2.1.2 Eliminación de duplicados

Cada imagen se inspeccionó para eliminar duplicados y garantizar que no hubiera repeticiones entre las fuentes. Se implementaron técnicas avanzadas de *hash* basadas en el contenido, permitiendo detectar similitudes visuales, incluso ante pequeñas variaciones o modificaciones, anexo 1. Adicionalmente, se realizó un proceso automatizado de etiquetado con un algoritmo especializado que facilitó la clasificación precisa y consistente del conjunto completo de imágenes.

5.2.1.3 Creación del *dataset* orgánicos e inorgánicos

El conjunto de datos final incluye 3,208 imágenes, divididas en 1,574 imágenes para la clase de residuos orgánicos y 1,634 imágenes para la clase de residuos inorgánicos, con un equilibrio proporcional entre ambas categorías. Este balance es fundamental para asegurar que los modelos de DL puedan generalizar adecuadamente, evitando sesgos hacia alguna categoría específica, y permitiendo que el entrenamiento sea representativo de escenarios reales, figura 2 y 3.

Figura 2

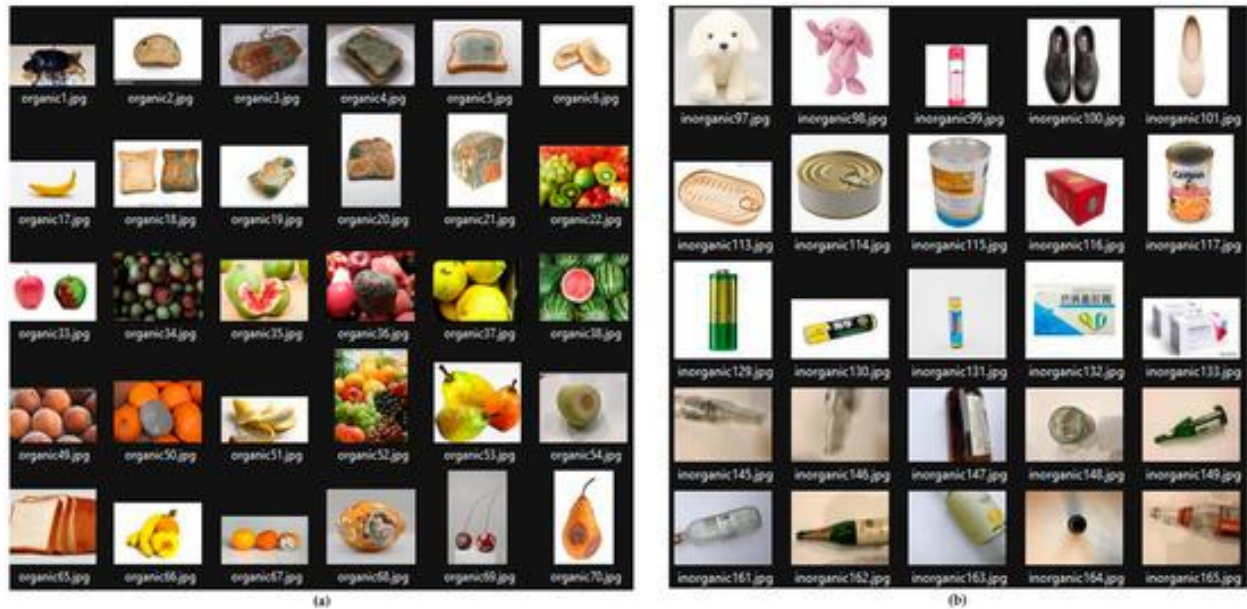
Muestra de las imágenes de los seis datasets utilizados, a: TrashNet, b: Garbage Huawei, c: Garbage classification data, d: Garbage dataset, e: Garbage Classification y f: Garbage Pictures for Classification.



Nota: Adaptado de R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, por (Castro-Bello, Romero-Juárez, Fuentes-Pacheco, Morales-Morales, et al., 2025), Sustainability 17.

Figura 3

Muestras del conjunto de datos para evaluar la red propuesta: a: residuos orgánicos y b: residuos inorgánicos.



Nota: Adaptado de R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, por (Castro-Bello, Romero-Juárez, Fuentes-Pacheco, Morales-Morales, et al., 2025), Sustainability 17.

5.2.2 Análisis de algoritmos CNN para la clasificación de RSU

5.2.2.1 Revisión sistemática de la literatura

Para conocer los distintos modelos de CNN se hizo una revisión de la literatura científica en bases de datos de Elsevier y JCR, considerando los últimos cinco años. La estrategia de búsqueda consistió en usar palabras claves como: “waste classification”, “Binary Waste Classification”, “Garbage Sorting CNN”, “Convolutional Neural Networks for Waste”, “Binary Classification CNN Models”.

5.2.3 Preparación y generación de datos para el entrenamiento del modelo

5.2.3.1 Conversión de imágenes al formato RGB

Para garantizar la uniformidad del procesamiento, las imágenes se convirtieron al formato RGB, independientemente de su formato original. Se realizó el preprocesamiento mediante un *script* de Python personalizado, este *script* procesó sistemáticamente todas las imágenes localizadas en el directorio fuente, cada imagen fue cargada utilizando la biblioteca *Pillow* (PIL), convirtiéndola al espacio de color estándar RGB (Rojo, Verde, Azul), y subsecuentemente guardada con su nombre de archivo y extensión originales en un directorio destino denominado *dataset_rgb/inorganico* y *dataset_rgb/orgánico*, anexo 2 y 3.

5.2.3.2 División del conjunto de datos en subconjuntos

Los conjuntos de datos de entrenamiento y validación se crearon a partir de la división de *trash_inf_train* en subconjuntos mediante la utilidad *train_test_split* de Scikit-learn. Considerando el 15% para validación y 85% para su entrenamiento, utilizando *random_state = 42* para asegurar la consistencia en la partición. La estratificación se realizó con base en la columna '*classification*', preservando la distribución original de las clases en ambos conjuntos resultantes, una práctica esencial para una evaluación fiable del modelo, anexo 4.

5.2.3.3 Aumento de imágenes en línea

El aumento de imágenes se realizó en línea durante el entrenamiento de la red utilizando la clase “*ImageDataGenerator*” de la biblioteca de redes neuronales de aprendizaje profundo Keras. Las técnicas de aumento aplicadas incluyeron reescalado, rotación, zoom, desplazamiento de ancho, de altura, volteo horizontal, volteo vertical y modo de relleno, tabla 2.

Tabla 2

Valores usados para el aumento de datos en fase de entrenamiento

ImageDataGenerator	
Parámetro	Valor
<i>rescale</i>	1./255
<i>rotation_range</i>	[0,5]
<i>zoom_range</i>	[0.0, 0.10]
<i>width_shift_range</i>	[0.0,0.1]
<i>height_shift_range</i>	[0.0,0.1]
<i>shear_range</i>	[0.0,0.5]
<i>horizontal_flip</i>	<i>True</i>
<i>vertical_flip</i>	<i>True</i>
<i>fill_mode</i>	<i>nearest</i>

5.2.4 Entrenamiento e identificación de redes residuales con la mayor extracción de características

5.2.4.1 Desarrollo de arquitectura basada en ResNet18

El desarrollo de la arquitectura para la clasificación de residuos se fundamentó en los principios de ResNet, tomando como referencia conceptual una estructura ResNet18, reconocida por su eficiente extracción de características y estabilidad en el entrenamiento de redes profundas gracias a sus conexiones de salto. Se optó por un diseño que, aunque inspirado en los bloques residuales básicos de ResNet18 (compuestos típicamente por dos capas convolucionales de 3×3), se enfocó en una reducción significativa de la complejidad. Esto se traduce, en una menor cantidad de filtros en la capa convolucional inicial y en las etapas subsiguientes, así como un número optimizado de repeticiones de bloques, diferenciándose marcadamente de las arquitecturas 'bottleneck' para lograr un modelo más ligero y adaptado a los recursos y datos disponibles para la tarea de clasificación binaria.

5.2.5 Experimentación y ajuste de la reducción de parámetros del modelo residual propuesto

5.2.5.1 Estructura de la red residual propuesta

Como resultado del proceso de experimentación y ajuste detallado, a continuación, se describen los componentes y configuración de la arquitectura residual modificada:

- **Capa de entrada**

Los datos de entrada consistieron en imágenes de 224×224 píxeles con tres canales de color (RGB). La red procesó estas imágenes mediante una serie de capas convolucionales que extraen características relevantes para su posterior clasificación de residuos en categorías orgánicas e inorgánicas.

- **Capa convolucional inicial**

La red se inició con una capa convolucional que utiliza un filtro de 7×7 , un *stride* de 1 y 16 filtros. Cuyo objetivo fue extraer características espaciales iniciales de las imágenes de entrada. Representando matemáticamente la operación de convolución bidimensional con la ecuación (1).

$$Y[i, j] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} X[i + m, j + n] \cdot W[m, n] + b \quad (1)$$

donde

$Y[i, j]$: valor de salida en la posición (i, j) después de la convolución

$X[i + m, j + n]$: valor de entrada en la posición (i, j)

$W[m, n]$: valor del filtro en la posición (m, n)

b : sesgo; M y N dimensiones del filtro

- **Normalización por lotes**

Tras la convolución, se aplicó la normalización ($\mu=0$, $\sigma=1$) por lotes antes de las funciones de activación de una capa para estabilizar el aprendizaje y acelerar la convergencia del modelo, para ajustar las activaciones de cada lote para que tengan una distribución normal, lo que ayuda a mitigar el problema del desplazamiento de covariables, ecuaciones (2), (3) y (4).

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \quad (2)$$

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (3)$$

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (4)$$

donde

x_i : valor de activación para la entrada

i, μ_B y σ_B^2 : media y la varianza del batch

ϵ : constante para evitar la división por cero

m : tamaño del batch

$\hat{x}_i$: valor normalizado

Se aplicó una transformación adicional para reescalar y desplazar los valores normalizados con los parámetros aprendibles γ y β , obteniendo y_i según la ecuación (5). Empleando estadísticas acumuladas durante la inferencia.

$$y_i = \gamma \hat{x}_i + \beta \quad (5)$$

- **Funciones de activación ReLU**

Finalmente, se aplicó una función de activación ReLU para introducir la no linealidad en el modelo, lo que permitió a la red aprender representaciones más complejas de los datos, ecuación (6).

$$ReLU(x) = \max(0, x) \quad (6)$$

Para cualquier entrada x , la función devuelve x si es > 0 , y 0 si $x \leq 0$. Matemáticamente, esto se puede expresar como la ecuación (7):

$$ReLU(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases} \quad (7)$$

Esta función es ampliamente utilizada en redes neuronales de aprendizaje profundo porque ayuda a abordar el problema de la desaparición del gradiente, lo que hace que el modelo sea más fácil de entrenar y, a menudo, logre un mejor rendimiento. Esta secuencia de operaciones (convolución, normalización por lotes y activación ReLU) permite a la red extraer y reforzar características significativas de las imágenes de entrada, facilitando la posterior clasificación de los residuos en categorías orgánicas e inorgánicas.

- **Bloques residuales**

Tras el bloque inicial, el modelo constó de cuatro bloques residuales. El primer bloque contiene 16 filtros, dos repeticiones y un paso; el segundo constó de 32 filtros, dos repeticiones y dos pasos; el tercero de 64 filtros, una repetición y dos pasos; y el cuarto de 128 filtros, una repetición y dos pasos. Estos bloques permitieron la creación de redes profundas sin problemas asociados a la desaparición de gradiente. Donde un bloque residual en la arquitectura R3sNet se definió condicionalmente, dependiendo si existe un cambio en la dimensionalidad de las características, ecuación (8).

$$y = \begin{cases} ReLU(F(x, \{W_i\}) + x) & \text{si no hay cambio de dimensionalidad} \\ ReLU(F(x, \{W_i\}) + W_s * x) & \text{si hay cambio de dimensionalidad} \end{cases} \quad (8)$$

donde

y : salida del bloque residual

x : entrada del bloque residual

$F(x, \{W_i\})$: representa las transformaciones en la rama principal, consistente en capas convolucionales seguidas de normalización por lotes y activación ReLU.

W_s : filtros de la convolución 1×1 utilizados en la conexión de atajo cuando se requiere ajustar la dimensionalidad.

$*$: operación de convolución

$ReLU$: función de activación aplicada tras la suma

Asegurando que, independientemente de los cambios en la dimensionalidad, la arquitectura mantiene una conexión de atajo efectiva que facilita el flujo de gradientes y estabiliza el entrenamiento de la red.

- **Bloques de identidad**

Los bloques de identidad son un subconjunto de bloques residuales que se utilizan cuando la dimensionalidad de la entrada y la salida del bloque no cambia. En este caso, la conexión de atajo es simplemente la identidad, es decir, la entrada se suma directamente a la salida de la rama principal sin ninguna transformación adicional, ecuación (9).

$$y = ReLU(F(x, \{W_i\}) + x) \quad (9)$$

donde

y : salida final del bloque de identidad

x : entrada al bloque de identidad

$(x, \{W_i\})$: transformaciones en la rama principal, consistentes en capas convolucionales seguidas de normalización por lotes y activación ReLU.

$ReLU$: función de activación aplicada tras la suma

- **Bloques de proyección**

Cuando el $stride > 1$, se utiliza un bloque de proyección para ajustar las dimensiones de la entrada y que estas coincidan con las dimensiones de la salida de la rama principal. Esto es esencial para garantizar la compatibilidad dimensional durante la suma de las ramas, ecuación (10).

$$y = ReLU(F(x, \{W_i\}) + Conv_{1 \times 1}(x, stride)) \quad (10)$$

donde

y : salida de los bloques de proyección

x : entrada al bloque residual

$F(x, \{W_i\})$: transformaciones en la rama principal, consistentes en capas convolucionales 3×3 seguidas de normalización por lotes y activación ReLU.

$Conv_{1 \times 1}(x, stride)$: operación de convolución 1×1 con $stride > 1$ para ajustar las dimensiones de x .

$ReLU$: función de activación aplicada tras la suma

Este bloque permite la resolución espacial y/o aumentar el número de filtros, ajustando las dimensiones de la conexión de atajo para mantener la consistencia dimensional.

- **Capa de agrupamiento global**

Después de aplicar varios bloques residuales, se utilizó una capa *Global Average Pooling* para reducir las dimensiones espaciales de la salida del último bloque residual. Esta operación calcula el promedio de las activaciones de cada canal, produciendo un vector de características de longitud C, donde C es el número de canales de salida, ecuación (11).

$$GAP(x) = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x[i, j, c] \quad (11)$$

donde

H y W : dimensiones espaciales de la salida del bloque residual

$x[i, j, c]$: valor de activación en la posición (i, j) y canal c

El resultado es un vector de características de dimensión C, donde cada elemento representa el promedio de activaciones de un canal específico. Esta capa transforma mapas de características espaciales en vectores de características globales, facilitando la conexión con la capa densa de salida para la clasificación.

- **Capa densa de salida**

La salida de la capa *Global Average Pooling* se conectó a una capa densa con una única neurona y activación sigmoide, la cual produce la probabilidad de clasificación binaria. Definiendo la operación en la capa densa, ecuación (12).

$$y = \sigma(W \cdot x + b) \quad (12)$$

donde

y : salida de la capa densa

x : vector de características resultante de la capa GAP

W : vector de pesos de la capa densa

b : sesgo

σ : función sigmoide, ecuación (13)

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (13)$$

La función sigmoide mapea cualquier valor real en un rango entre 0 y 1, interpretándose como la probabilidad de pertenecer a una de las clases (orgánicos vs inorgánicos). Al usar una sola neurona en la capa de salida con una función de activación sigmoide, el modelo reduce el número de parámetros necesarios, lo que resulta en un entrenamiento potencialmente más rápido con menor riesgo de sobreajuste. Además, un único valor de salida indica directamente la probabilidad p de la clase positiva, y su complemento $1 - p$ corresponde a la clase negativa, por lo que no se requiere una segunda neurona en este caso.

- **Función de pérdida y optimización**

Para entrenar la red, se utilizó *Binary Focal Crossentropy* como función de pérdida, la cual es especialmente eficaz en problemas con clases desequilibradas. Esta elección se fundamenta en la capacidad para manejar la desproporción entre clases, enfocándose en los ejemplos más difíciles y reduciendo la influencia de los ejemplos bien clasificados, ecuación (14).

$$FL(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t) \quad (14)$$

donde

α : factor de ajuste para compensar el desbalance entre clases

p_t : probabilidad predicha para la clase verdadera

γ : factor de modulación que reduce la contribución de los ejemplos bien clasificados y enfoca el aprendizaje en los ejemplos difíciles.

Estos hiperparámetros han sido seleccionados basándose en estudios previos que demuestran su eficacia en tareas de clasificación binaria desbalanceada (Lin et al., 2017). La combinación de $\alpha = 0.25$ y $\gamma = 2.0$ permite al modelo mejorar su rendimiento en la detección de residuos de la clase minoritaria, evitando que se sesgue hacia la clase mayoritaria.

El modelo es optimizado utilizando el *Stochastic Gradient Descent* (SGD) con un momentum de 0.9 y una tasa de aprendizaje ajustable. La actualización de los pesos se realiza mediante la ecuación (15) y (16).

$$v_t = \beta v_{t-1} + (1 - \beta) \nabla L(\theta) \quad (15)$$

$$\theta_t = \theta_{t-1} - \eta v_t \quad (16)$$

donde

v_t : velocidad en el tiempo

β : factor de momentum, ayuda a acelerar la convergencia en direcciones relevantes y suavizar las oscilaciones durante el entrenamiento.

$\nabla L(\theta)$: gradiente de la función de pérdida con respecto a los pesos θ

η : tasa de aprendizaje que determina el tamaño de los pasos de actualización

Los hiperparámetros del optimizador SGD se seleccionaron y ajustaron según recomendaciones ampliamente aceptadas en la literatura sobre entrenamiento de redes neuronales profundas. El momentum se estableció en 0.9, puesto que este parámetro facilita una convergencia más rápida y suaviza posibles oscilaciones al acumular información previa sobre la dirección del gradiente. Se activó la variante de momentum de Nesterov, que permite una anticipación más precisa de las direcciones del gradiente. La tasa de aprendizaje inicial se estableció en 0.01, lo que proporciona una reducción gradual y estable del tamaño de paso a medida que avanza el entrenamiento. Finalmente, para mejorar aún más la estabilidad y la eficiencia del proceso de aprendizaje, se utilizó ReduceLROnPlateau que ajusta dinámicamente la tasa de aprendizaje si la pérdida de validación no mejora después de varias épocas consecutivas.

5.2.6 Evaluación del algoritmo de red residual

5.2.6.1 Análisis de parámetros

La evaluación del rendimiento se basó en la precisión, la matriz de confusión, la recuperación, la puntuación F1, la precisión y la pérdida de entropía cruzada observada durante el entrenamiento, la validación y las pruebas del modelo.

- **Exactitud**

Representa la proporción de clasificaciones correctas realizadas por el modelo de clasificación de residuos y se puede definir mediante la siguiente fórmula:

$$Exactitud = \frac{VP + VN}{VP + FN + VN + FP}$$

donde Verdadero Positivo (VP) se refiere al número de imágenes con elementos orgánicos que se predijeron correctamente, Verdadero Negativo (VN) indica el número de imágenes de desechos inorgánicos que se predijeron correctamente, Falso Positivo (FP) indica el número de imágenes de

desechos inorgánicos que se predijeron como pertenecientes a la clase orgánica y Falso Negativo (FN) indica el número de imágenes de clase orgánica que se predijeron como inorgánicas.

- **Precisión**

Es la proporción de muestras genuinamente positivas entre todas las muestras previstas como positivas y puede formularse de la siguiente manera:

$$Precisión = \frac{VP}{VP + FP}$$

- **Sensibilidad**

Proporción de todas las muestras positivas que se predicen correctamente como positivas y se puede expresar de la siguiente manera:

$$Sensibilidad = \frac{VP}{VP + FN}$$

- **Métrica F1**

Media armónica de precisión y recuperación, donde una puntuación F1 alcanza su mejor valor en 1 y su peor valor en 0:

$$Métrica F1 = \frac{2 \times VP}{2 \times VP + FP + FN}$$

Capítulo VI. Resultados

6.1 *Dataset* de RSU

El conjunto de datos se conforma de 3,208 imágenes en formato JPG, divididas en 1,574 imágenes para la clase de residuos orgánicos y 1,634 de inorgánicos con un equilibrio proporcional en ambas categorías, figura 4 y 5. Disponible en <https://github.com/victorMRJDev/datasetR3sNet>.

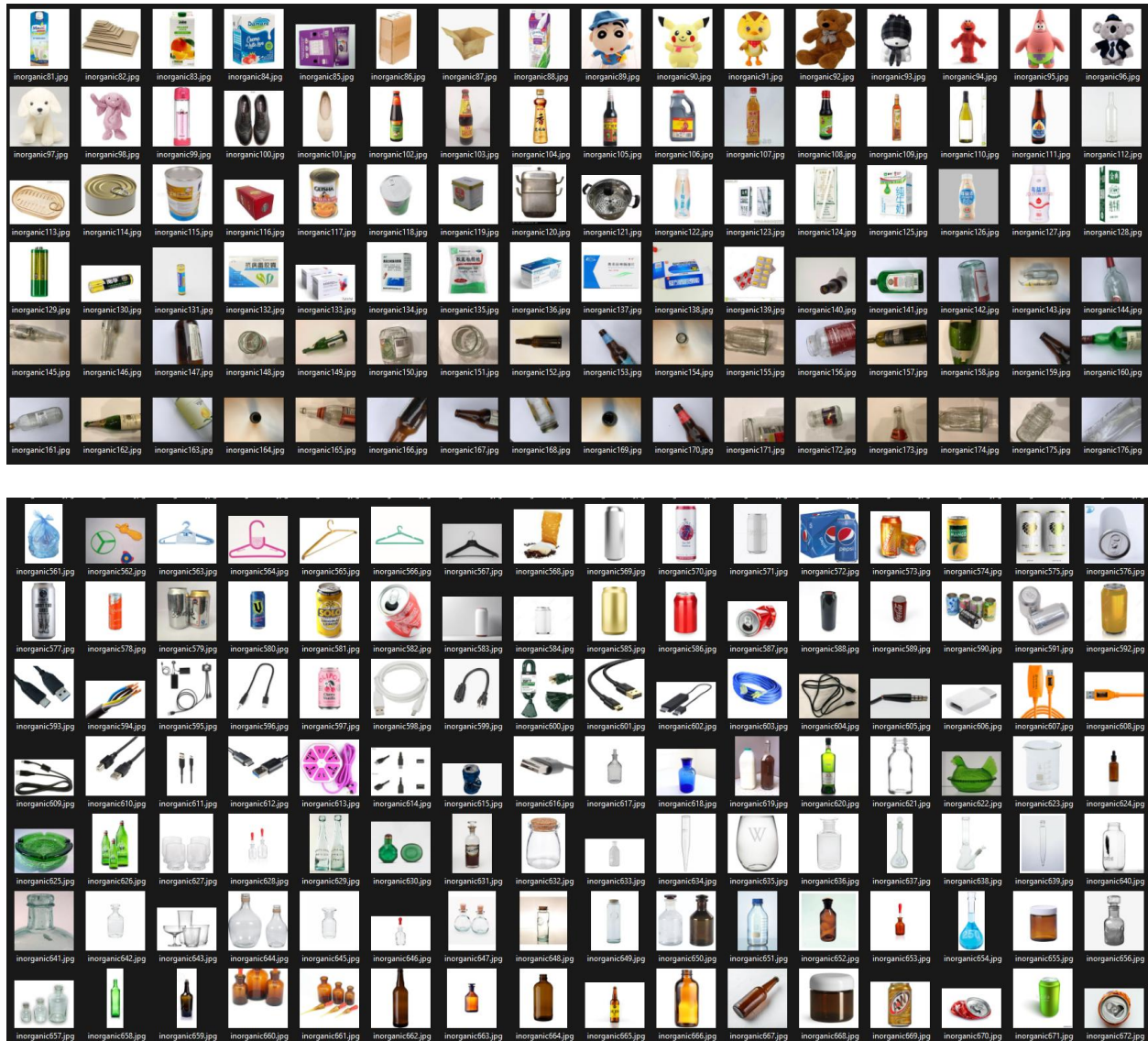
Figura 4

Dataset RSU orgánicos.



Figura 5

Dataset RSU inorgánicos.



6.2 Algoritmos CNN para la clasificación de RSU

Como resultado de la revisión de la literatura, se hizo un análisis de las arquitecturas de los modelos CNN en función de sus parámetros, GFLOPS, ventajas y desventajas en la clasificación de RSU, tabla 3.

Tabla 3

Análisis de las diferentes arquitecturas aplicadas en la clasificación de RSU.

Modelo	Parámetros (Millones)	GFLOPS	Ventajas	Desventajas
ResNet50	25.6	4.09	• Conexiones residuales que facilitan el entrenamiento de redes muy profundas, mitigando el gradiente desvanecido. • Buen rendimiento de clasificación (históricamente fuerte en ImageNet).	• En conjuntos de datos pequeños puede sobreajustar.
VGG16	138.4	15.47	• Arquitectura simple y homogénea (filtros 3×3), fácil de entender e implementar.	• Peor desempeño global en tareas modernas comparada con redes posteriores • Sin conexiones residuales, sufre más de gradiente desvanecido/explosivo.
DenseNet 121	8.0	2.83	• Conectividad densa, reutiliza mapas de características en capas posteriores, mejorando el flujo de gradientes y la eficiencia de parámetros	• Uso intensivo de memoria: las conexiones densas (todas-a-todas) incrementan el uso de memoria en entrenamiento/inferencia. Puede resultar en inferencia

				más lenta en GPUs con poca memoria.	
EfficientNet-B0	5.3	0.39	• Escalado compuesto (profundidad, ancho y resolución) optimiza el balance precisión/eficiencia. • Óptimo desempeño general: mejor precisión con la menor cantidad de parámetros entre redes comparadas	• Arquitectura compleja derivada de AutoML. • Puede requerir optimizaciones específicas para hardware, como GPUs o TPUs.	

6.3 Preparación y generación de datos para el entrenamiento del modelo CNN

El conjunto de datos se dividió en entrenamiento y validación (proporción 85:15), resultando en 2,730 imágenes para entrenamiento y 482 para validación, anexo 5. El conjunto de prueba se mantuvo independiente, figura 6 y 7.

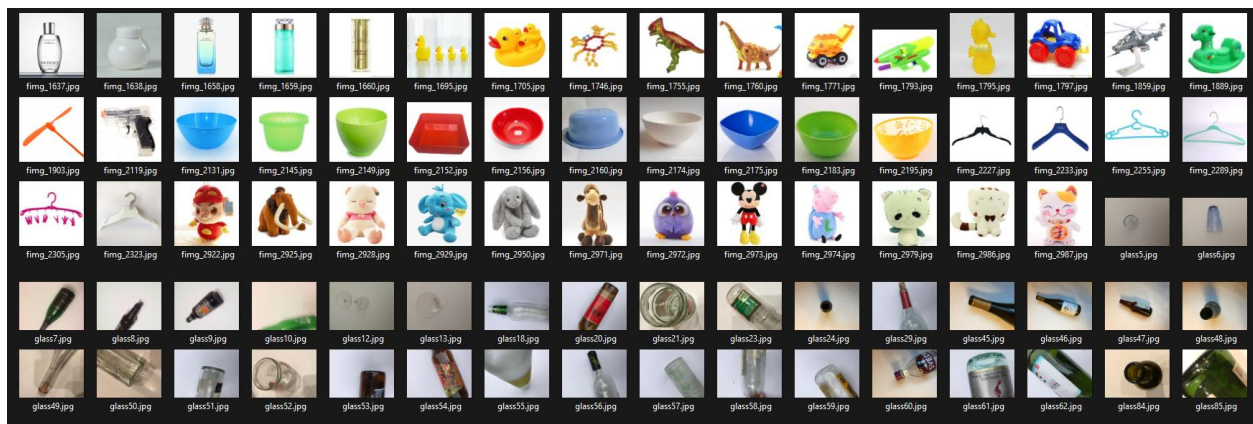
Figura 6

Dataset de prueba orgánicos.



Figura 7

Dataset de prueba inorgánicos.



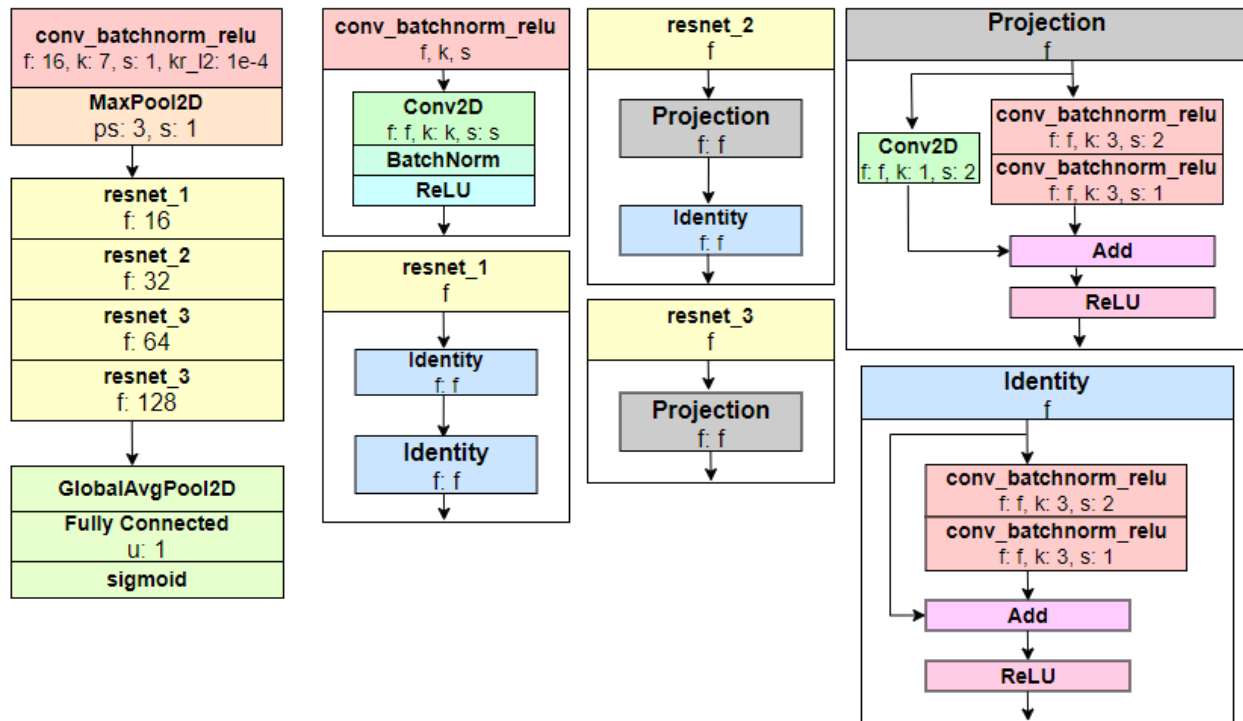
El aumento de datos en línea se realizó en función de los parámetros rescale, rotación, zoom, volteo horizontal, vertical y relleno, anexo 6.

6.4 Entrenamiento e identificación de redes residuales

La arquitectura propuesta se configuró en función de ResNet18, obteniendo una versión ligera, figura 8.

Figura 8

Diagrama de arquitectura R3sNet



Nota: Adaptado de R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, por (Castro-Bello, Romero-Juárez, Fuentes-Pacheco, Morales-Morales, et al., 2025), Sustainability 17.

6.5 Experimentación y ajustes de parámetros

La experimentación y ajustes de parámetros, consistió en 10 repeticiones de entrenamiento para los modelos R3sNet y ResNet50 preentrenados utilizando los mismos parámetros, tabla 4. Obteniendo un total de 334,369 parámetros entrenables, figura 9. Una reducción del 98.60% en los hiperparámetros y 65.17% en GFLOPS, logrando un modelo compacto con un peso de 2.82 MB, tabla 5.

Tabla 4

Hiperparámetros utilizados en R3sNet.

Hiperparámetro	Valor
Input	224, 224, 3
Batch	16
Activation	Sigmoid
Optimizer	SGD
learning_rate	1×10^{-3}
Epochs	30

Figura 9

Cantidad de parámetros de R3sNet.

Layer (type)	Output Shape	Param #	Connected to
=====			
input_1 (InputLayer)	[(None, 224, 224, 3 0)]		[]
conv2d (Conv2D)	(None, 224, 224, 16 2368)		['input_1[0][0]']
batch_normalization (BatchNorm alization)	(None, 224, 224, 16 64)		['conv2d[0][0]']
re_lu (ReLU)	(None, 224, 224, 16 0)		['batch_normalization[0][0]']
max_pooling2d (MaxPooling2D)	(None, 222, 222, 16 0)		['re_lu[0][0]']
conv2d_1 (Conv2D)	(None, 222, 222, 16 2320)		['max_pooling2d[0][0]']
...			
Total params: 334,369			
Trainable params: 333,185			
Non-trainable params: 1,184			

Tabla 5

Comparación de parámetros y GFLOPS entre R3sNet y ResNet50.

Modelo	Parámetros	GFLOPS
R3sNet	334, 369	43.2031
ResNet50	23,587,712	124.0152

6.6 Evaluación del algoritmo de red residual

En la evaluación del algoritmo de la red residual, las métricas de rendimiento estimado incluyeron precisión, sensibilidad, Métrica F1 y exactitud. También se obtuvieron las curvas de pérdida y exactitud, el modelo convergió en la sexta época, donde las métricas de pérdida y exactitud se estabilizaron. Aunque se produjeron pequeñas fluctuaciones, posteriormente, estas fueron insignificantes, lo que sugiere que el modelo había alcanzado un punto de estabilidad. Por tanto, el entrenamiento se detuvo después de 30 épocas. Los resultados obtenidos en el conjunto de pruebas presentan la media y desviación estándar, tabla 6.

Tabla 6

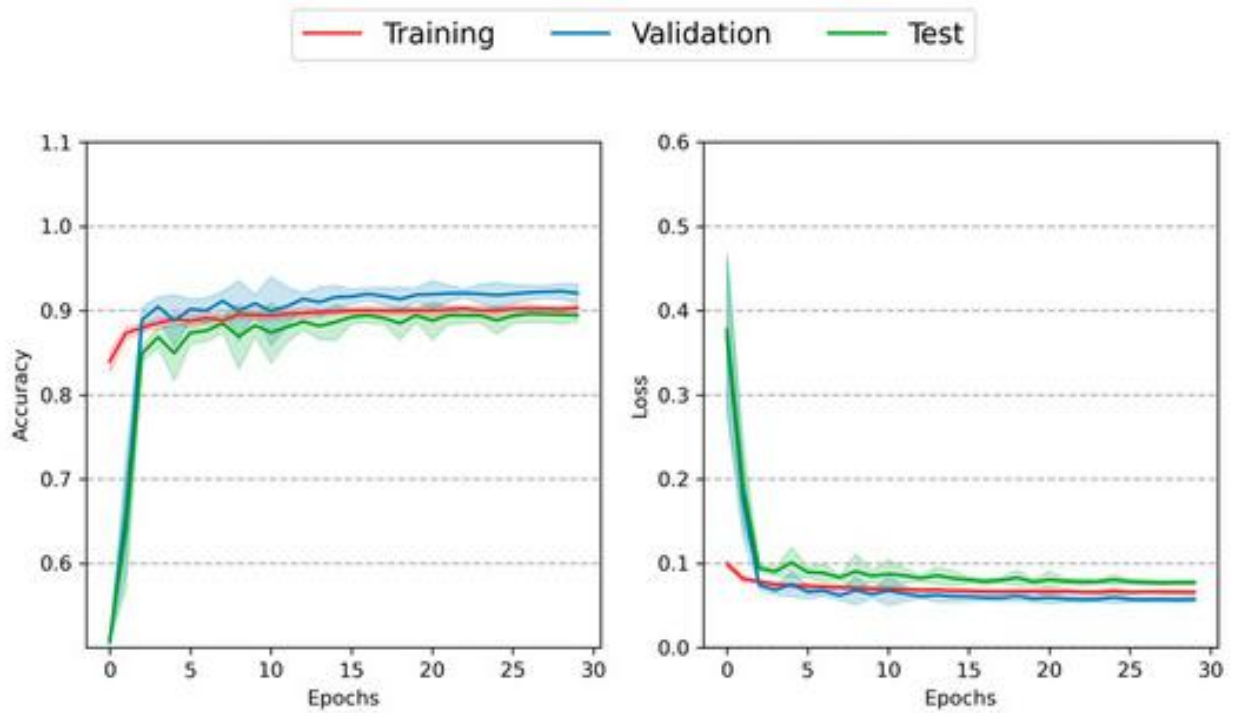
Métricas de rendimiento para R3sNet y ResNet50

Dataset	Métrica	R3sNet	ResNet50 Preentrenado
Train	Accuracy	0.8941 ± 0.0053	0.8026 ± 0.0109
	Loss	0.0701 ± 0.0027	0.4281 ± 0.0170
Validation	Accuracy	0.8901 ± 0.0156	0.8128 ± 0.0162
	Loss	0.0758 ± 0.0102	0.4267 ± 0.0211
Test	Accuracy	0.8633 ± 0.156	0.8167 ± 0.0115
	Loss	0.0965 ± 0.0109	0.4011 ± 0.0173

La media y desviación estándar en la validación y pruebas durante el entrenamiento de R3sNet mostró un desempeño superior al de ResNet50, figura 10 y 11.

Figura 10

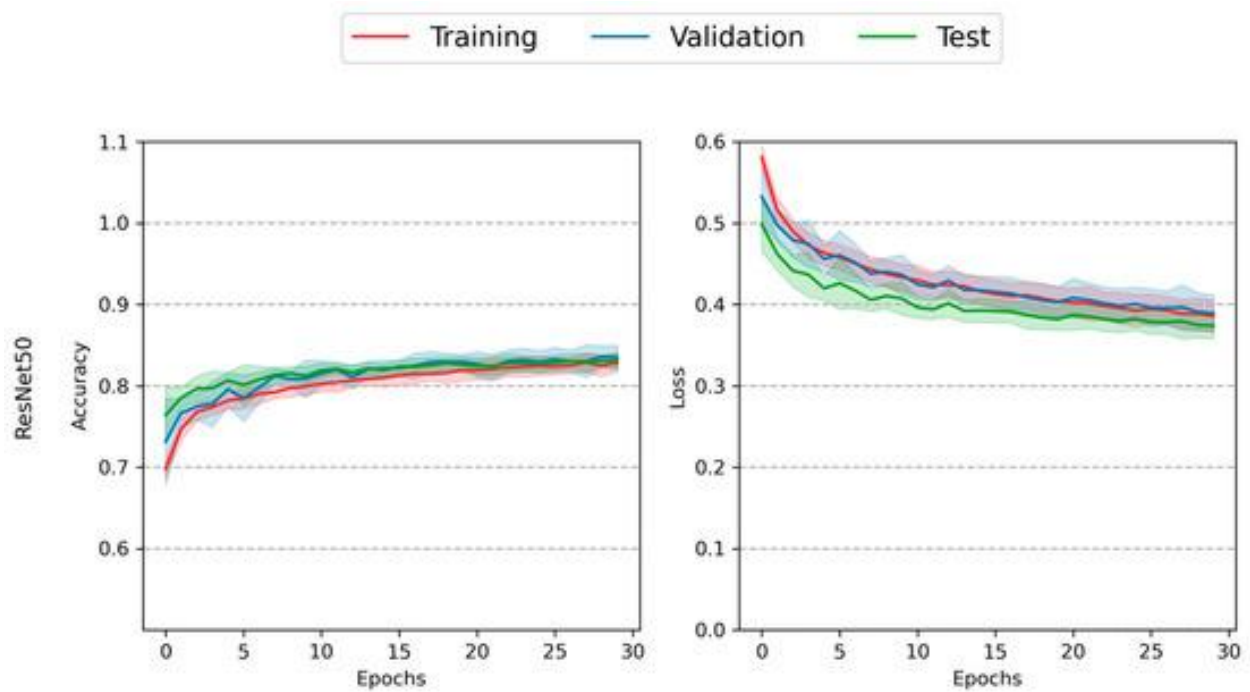
Gráfico que muestra la media y la desviación estándar de R3sNet en 10 ejecuciones.



Nota: Adaptado de R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, por (Castro-Bello et al., 2025), Sustainability 17.

Figura 11

Gráfico de la media y la desviación estándar utilizando ResNet50 preentrenado en 10. ejecuciones.

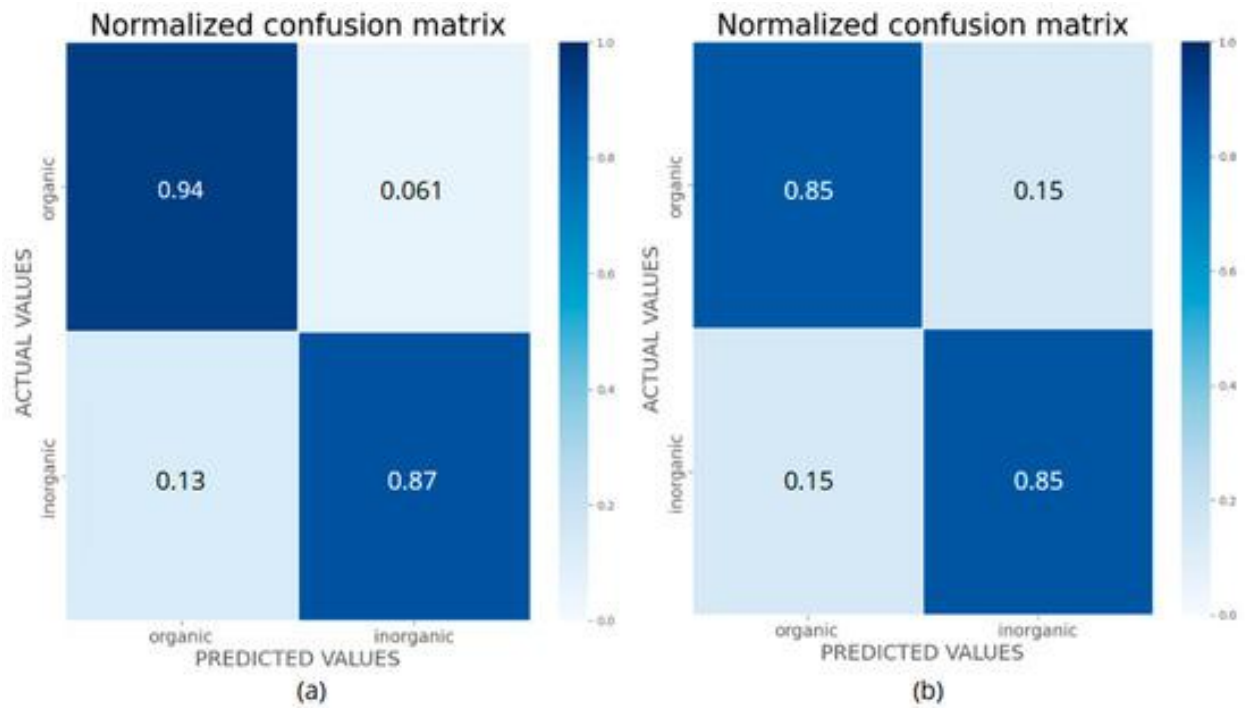


Nota: Adaptado de R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, por (Castro-Bello et al., 2025), Sustainability 17.

Para garantizar la robustez y la fiabilidad de los resultados, se realizaron 10 repeticiones con imágenes aleatorias de los conjuntos de entrenamiento y pruebas, figura 12.

Figura 12

Matriz de confusión de los modelos (a) R3sNet y (b) ResNet50 preentrenado.



Nota: Adaptado de R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, por (Castro-Bello et al., 2025), Sustainability 17.

Capítulo VII. Discusión

La clasificación de RSU ha evolucionado significativamente gracias a los avances en las CNN. Este estudio implementó una versión optimizada de ResNet, denominada R3sNet, diseñada específicamente para abordar la clasificación binaria de residuos con eficiencia computacional y alta precisión. Se destacan varios aspectos relevantes al comparar los resultados obtenidos con investigaciones relacionadas. En términos de rendimiento, R3sNet alcanzó una precisión del 87% para la clasificación de residuos orgánicos y del 94% para residuos inorgánicos, superando los resultados obtenidos por los modelos CNN convencionales implementados por Nnamoko et al. (2022), que promedió una precisión del 80.88% en ambas clasificaciones utilizando una arquitectura ligera de cinco capas con imágenes de menor resolución. Por otro lado, Yang et al. (2022) implementaron una CNN pequeña que alcanzó un 96.77% en su conjunto de datos personalizado y un 93.72% en el conjunto TrashNet. Emplearon técnicas avanzadas de preprocesamiento, como algoritmos de brillo adaptativo y operadores Canny para el recorte de fondo, lo que podría haber contribuido a una mayor precisión. En comparación, R3sNet, al integrar bloques residuales con conexiones de acceso directo, funciones de activación ReLU y normalización por lotes, demuestra ser una solución eficiente que mitiga problemas comunes como la desaparición de gradientes y el sobreajuste. Además, la función de pérdida de entropía cruzada focal binaria gestionó eficazmente el desequilibrio de clases. En términos de eficiencia computacional, R3sNet busca equilibrar el rendimiento. A diferencia de modelos como DenseNet121 entrenados para MSW (Mao et al., 2021; Zhang, et al., 2021) con una gran cantidad de parámetros (más de ocho millones), R3sNet reduce significativamente los requisitos computacionales, manteniendo una precisión competitiva. Esto resulta especialmente útil para implementaciones en entornos con recursos limitados, como los dispositivos IoT. Nnamoko et al.

(2022) utilizó un conjunto de datos tomado de Sekar (2018) que contiene 25,077 imágenes, uno de los conjuntos de datos utilizados en este estudio. Sin embargo, se identificaron y eliminaron imágenes duplicadas durante la fase de selección del conjunto de datos. Este paso es crucial para evitar sesgos de entrenamiento y garantizar que el cálculo del rendimiento del modelo sea representativo y fiable. Además, una evaluación consistente con una combinación de datos específica permitió obtener resultados más robustos y generalizables. Aunque el volumen de datos utilizado es relativamente limitado, se consideró suficiente para entrenar eficazmente el modelo R3sNet gracias a las estrategias implementadas para maximizar la diversidad de muestras. Las técnicas de aumento de datos en línea y la eliminación de duplicados mejoraron la variabilidad del conjunto de entrenamiento sin requerir la recopilación de datos adicionales. Tales estrategias aumentaron la atractiva diversidad de ejemplos presentados al modelo, mitigando las limitaciones derivadas de un menor número de muestras y reduciendo el riesgo de sobreajuste. El aumento de datos en línea aplica nuevas transformaciones a las imágenes originales, generando variaciones en cada época de entrenamiento y garantizando imágenes diferentes en cada iteración. De este modo, R3sNet capturó un conjunto más robusto de características generales. Un conjunto de datos más extenso podría mejorar aún más el rendimiento y la generalización del modelo, puesto que un mayor número de ejemplos tiende a cubrir la variabilidad de los residuos de forma más compleja. Sin embargo, los resultados indican que el modelo logró un rendimiento competitivo a pesar de su tamaño limitado. Esto sugiere que la cantidad de datos disponibles, enriquecida mediante el aumento de datos y la limpieza de duplicados, fue adecuada para entrenar eficazmente R3sNet, logrando un equilibrio satisfactorio entre la complejidad del modelo y la información proporcionada.

Las arquitecturas convolucionales tradicionales con muchos parámetros suelen sobreajustarse cuando se dispone de acceso a conjuntos de datos limitados. Los conjuntos de datos públicos para la clasificación de MSW suelen ser relativamente pequeños; por ejemplo, se suelen disponer de unas 25 000 imágenes para tareas binarias como la diferenciación entre orgánicos e inorgánicos. Esto contrasta marcadamente con conjuntos de datos extensos como ImageNet. En estas situaciones, entrenar redes con millones de parámetros desde cero puede provocar que los modelos memoricen los datos de entrenamiento en lugar de aprender patrones generalizables. Si bien los modelos convencionales como VGG-16, ResNet50 o Dense-Net-121 han demostrado una alta precisión en otras tareas, comprenden muchas capas y parámetros (VGG-16 contiene aproximadamente 138 millones de parámetros) y requieren una potencia de cálculo considerable. Por consiguiente, no son adecuados para aplicaciones en entornos con recursos limitados, como los dispositivos integrados. Incluso los intentos de reducir o congelar capas en modelos preentrenados suelen generar modelos voluminosos que afectan negativamente al almacenamiento y la eficiencia operativa.

Una limitación del presente estudio es que el modelo se probó únicamente en dos clases (orgánica e inorgánica), lo que podría no reflejar plenamente su potencial en un escenario multiclase. Por lo tanto, se planea ampliar el análisis a un problema multiclase, lo que podría requerir ajustes en la arquitectura del modelo y el ajuste de hiperparámetros para optimizar su rendimiento en tareas más complejas.

R3sNet exhibe una precisión excepcional, con un rápido aprendizaje inicial, estabilización durante las épocas finales y sólidas capacidades de generalización. La arquitectura optimizada y la configuración de entrenamiento adecuada mejoran estos resultados, convirtiéndolo en una opción práctica para tareas de clasificación de MSW.

Los resultados demuestran el potencial de R3sNet para una clasificación eficiente y precisa de RSU. Sin embargo, futuras investigaciones podrían centrarse en la validación del modelo con conjuntos de datos más amplios y diversos, así como en la incorporación de técnicas avanzadas de preprocesamiento para mejorar su rendimiento. Por ejemplo, analizando residuos en imágenes con fondos complejos y condiciones de iluminación variables mediante el aprendizaje por transferencia con nuestra red preentrenada.

R3sNet tiene un gran potencial para la clasificación de residuos en el mundo real. Su diseño ligero permite integrarlo en sistemas de clasificación automática (bandas transportadoras o máquinas de reciclaje), reduciendo costes y mejorando la eficiencia de la separación de residuos. Esto podría reducir la cantidad de residuos en los vertederos y contribuir a los objetivos de la economía circular, especialmente en regiones con recursos limitados. Sin embargo, desafíos como los costes iniciales de hardware (cámaras y otros sensores) y las variaciones ambientales podrían afectar su rendimiento, requiriendo optimizaciones como la recopilación de datos en diversas condiciones.

En comparación con estudios recientes, R3sNet se alinea con la tendencia hacia modelos más eficientes. Por ejemplo, el trabajo de Girsang et al. (2023) usaron MobileNet con aprendizaje por transferencia y se logró una precisión del 93.35% en 2400 imágenes. Nuestro modelo muestra una precisión menor, posiblemente debido a la falta de aprendizaje por transferencia, destacando la eficacia de reutilizar pesos previamente entrenados, lo que sugiere que R3sNet podría beneficiarse de esta técnica para mejorar su rendimiento, especialmente con conjuntos de datos pequeños.

El rendimiento de nuestro modelo es ligeramente inferior al de Chhabra et al. (2024) quienes lograron una precisión del 93.28 % en la clasificación de residuos orgánicos y reciclables utilizando una arquitectura de red neuronal convolucional mejorada. Esta diferencia podría deberse a las distintas estrategias de optimización y regularización empleadas. Mientras Chhabra et al.

(2024) utilizó capas convolucionales profundas, LeakyReLU y dropout, nuestro enfoque no incorpora estas mejoras, lo que limita la extracción de características. Además, la mayor cantidad y calidad de datos que utilizaron (25 077 imágenes, 70 % de entrenamiento, 30 % de prueba) sugiere que un conjunto de datos más grande o equilibrado podría mejorar el rendimiento de nuestra arquitectura.

La incorporación de modificaciones como LeakyReLU o tasas de abandono ajustadas podría beneficiar nuestra propuesta, especialmente en escenarios con alta variabilidad de residuos. Si bien nuestra precisión es ligeramente inferior, R3sNet está optimizado para la eficiencia con significativamente menos parámetros, lo que lo hace adecuado para entornos con recursos computacionales limitados. En conclusión, R3sNet ofrece una solución eficiente y escalable para la clasificación binaria de RSU, superando a modelos más pesados como ResNet50 y siendo competitivo frente a enfoques recientes. Mientras que la DCNN mejorada de Chhabra et al. (2024) prioriza la precisión en contextos más exigentes para aplicaciones del mundo real, R3sNet es ideal en entornos de bajo coste y DCNN podría ser la opción preferida donde la reducción de errores es una prioridad. Las investigaciones futuras podrían combinar la eficiencia de R3sNet con la precisión de modelos como DCNN, optimizando su implementación en sistemas de gestión de residuos en condiciones variables.

Capítulo VIII. Conclusiones

Se presenta una arquitectura optimizada basada en ResNet, denominada R3sNet, diseñada específicamente para la clasificación binaria de RSU. Los resultados destacan el alto rendimiento del modelo propuesto, con una precisión del 87% en la clasificación de residuos orgánicos y del 94% en la de residuos inorgánicos. Estas métricas superan las de ResNet50 y demuestran que una arquitectura relativamente pequeña, que incluye bloques residuales, funciones de activación ReLU, normalización por lotes y la función de pérdida de entropía cruzada focal binaria, contribuye significativamente al rendimiento y la estabilidad del modelo.

La arquitectura R3sNet ofrece una precisión competitiva y optimiza el uso de recursos computacionales, lo que la hace ideal para su implementación en entornos con limitaciones de hardware, como dispositivos IoT. Además, su diseño facilita la generalización y minimiza problemas comunes en redes profundas, como el sobreajuste y desvanecimiento del gradiente. En comparación con ResNet50, R3sNet reduce los hiperparámetros en un 98.60% y los GFLOPS en un 65.17%.

Es importante destacar que el diseño eficiente de R3sNet contribuye a la sostenibilidad de varias maneras. En primer lugar, reduce la carga en los vertederos, promueve una economía circular y permite una clasificación de residuos más eficaz. En segundo lugar, la menor demanda computacional se refleja en el consumo energético, lo que lo hace adecuado para su implementación en entornos con limitaciones de hardware, como los dispositivos IoT utilizados en aplicaciones de ciudades inteligentes. Estas contribuciones indican que la investigación contribuye a los ODS 11, 12 y 13, lo que refuerza el potencial del modelo para el desarrollo urbano sostenible.

En conclusión, R3sNet supone un avance significativo en la gestión inteligente de los RSU. Ofrece una solución a los retos que surgen, garantizando que el desarrollo tecnológico vaya de la mano con el desarrollo sostenible.

Referencias bibliográficas

1. Azis, F. A., Suhaimi, H., & Abas, E. (2020). Waste Classification using Convolutional Neural Network. *Proceedings of the 2020 2nd International Conference on Information Technology and Computer Communications*, 9–13. <https://doi.org/10.1145/3417473.3417474>
2. Beciéz Salazar, F. A. (2020). *Desarrollo de una aplicación para la identificación de PET, cartón y papel por medio de una red convolutiva*. <https://rinacional.tecnm.mx/jspui/handle/TecNM/2688>
3. Bhattacharya, S., Kumar, A., Krishav, K., Panda, S., Vidhyapathi, C. M., Sundar, S., & Karthikeyan, B. (2024). Self-Adaptive Waste Management System: Utilizing Convolutional Neural Networks for Real-Time Classification. *Engineering Proceedings*, 62(1). <https://doi.org/10.3390/engproc2024062005>
4. Castro-Bello, M., Romero-Juárez, V., Fuentes-Pacheco, J., Zagal-Barrera, S. R., Bárcenas-Nava, A., & Gutiérrez-Valencia, D. E. (2025). Dataset de clasificación de Residuos Sólidos Urbanos para Redes Neuronales Convolucionales. *Dilemas Contemporáneos: Educación, Política y Valores*. <https://doi.org/10.46377/DILEMAS.V12I3.4675>
5. Castro-Bello, M., Romero-Juárez, V. M., Fuentes-Pacheco, J., Morales-Morales, C., Marmolejo-Vega, C. V., Zagal-Barrera, S. R., Gutiérrez-Valencia, D. E., & Marmolejo-Duarte, C. (2025). R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images. *Sustainability* 2025, Vol. 17, Page 3502, 17(8), 3502. <https://doi.org/10.3390/SU17083502>
6. Chhabra, M., Sharan, B., Elbarachi, M., & Kumar, M. (2024). Intelligent waste classification approach based on improved multi-layered convolutional neural network.

Multimedia Tools and Applications, 83(36), 84095–84120. <https://doi.org/10.1007/s11042-024-18939-w>

7. Chollet, F. (2021). *Deep Learning with Python* (Second). Manning. https://www.manning.com/books/deep-learning-with-python-second-edition?a_aid=keras
8. Girsang, A. S., Pratama, H., & Agustinus, L. P. S. (2023). Classification Organic and Inorganic Waste with Convolutional Neural Network Using Deep Learning. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2), 343–348. <https://www.ijisae.org/index.php/IJISAE/article/view/2637>
9. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
10. Han, S.-H., Kim, K. W., Kim, S., & Youn, Y. C. (2018). Artificial Neural Network: Understanding the Basic Concepts without Mathematics. *Dement Neurocogn Disord*, 17(3), 83–89. <https://doi.org/10.12779/dnd.2018.17.3.83>
11. He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. <https://arxiv.org/abs/1512.03385>
12. <https://www.kaggle.com/datasets/mascot9183/garbage-pictures-for-classification>. (2022). *Garbage Pictures for Classification*. <https://www.kaggle.com/datasets/mascot9183/garbage-pictures-for-classification?select=Garbage+classification>
13. Islam, Md. S. Bin, Sumon, Md. S. I., Majid, M. E., Abul Kashem, S. Bin, Nashbat, M., Ashraf, A., Khandakar, A., Kunju, A. K. A., Hasan-Zia, M., & Chowdhury, M. E. H. (2025). ECCDN-Net: A deep learning-based technique for efficient organic and recyclable waste classification. *Waste Management*, 193, 363–375. <https://doi.org/https://doi.org/10.1016/j.wasman.2024.12.023>

14. Kaza, S., Yao, L. C., Bhada-Tata, P., & Van Woerden, F. (2018). What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. <https://doi.org/10.1596/978-1-4648-1329-0>
15. Kunwar, S. (2024). *Garbage Dataset*. Kaggle. <https://doi.org/10.34740/KAGGLE/DSV/10182596>
16. Lapo, J. R., & Cumbicus-Pineda, O. M. (2024). Detection of recyclable solid waste using convolutional neural networks and PyTorch. *IEEE Latin America Transactions*, 22(6), 475–483. <https://doi.org/10.1109/TLA.2024.10534304>
17. LGEEPA. (2024). *LEY GENERAL DEL EQUILIBRIO ECOLÓGICO Y LA PROTECCIÓN AL AMBIENTE*.
18. LGPGIR. (2023). *LEY GENERAL PARA LA PREVENCIÓN Y GESTIÓN INTEGRAL DE LOS RESIDUOS*.
19. Lin, K., Zhao, Y., Gao, X., Zhang, M., Zhao, C., Peng, L., Zhang, Q., & Zhou, T. (2022). Applying a deep residual network coupling with transfer learning for recyclable waste sorting. *Environmental Science and Pollution Research*, 29(60), 91081–91095. <https://doi.org/10.1007/s11356-022-22167-w>
20. Lin, K., Zhao, Y., Kuo, J.-H., Deng, H., Cui, F., Zhang, Z., Zhang, M., Zhao, C., Gao, X., Zhou, T., & Wang, T. (2022). Toward smarter management and recovery of municipal solid waste: A critical review on deep learning approaches. *Journal of Cleaner Production*, 346, 130943. <https://doi.org/https://doi.org/10.1016/j.jclepro.2022.130943>
21. Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal Loss for Dense Object Detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007. <https://doi.org/10.1109/ICCV.2017.324>

22. Malik, M., Sharma, S., Uddin, M., Chen, C.-L., Wu, C.-M., Soni, P., & Chaudhary, S. (2022). Waste Classification for Sustainable Development Using Image Recognition with Deep Learning Neural Network Models. *Sustainability*, 14(12). <https://doi.org/10.3390/su14127222>
23. Mao, W.-L., Chen, W.-C., Wang, C.-T., & Lin, Y.-H. (2021). Recycling waste classification using optimized convolutional neural network. *Resources, Conservation and Recycling*, 164, 105132. <https://doi.org/10.1016/j.resconrec.2020.105132>
24. Mohamed, M. (2020). *Garbage Classification (12 classes)*. <https://www.kaggle.com/datasets/mostafaabla/garbage-classification>
25. Nnamoko, N., Barrowclough, J., & Procter, J. (2022). Solid Waste Image Classification Using Deep Convolutional Neural Network. *Infrastructures*, 7(4). <https://doi.org/10.3390/infrastructures7040047>
26. ONU-HABITAT. (2018, October 1). *ONU-Habitat - Día Mundial del Hábitat 2018*. <https://onuhabitat.org.mx/index.php/dia-mundial-del-habitat-2018>
27. Rondon Toro, E., Szanto Narea, M., Pacheco, J. F., Contreras, E., & Galvez, A. (2016). Guia General para la gestion de residuos solidos domiciliarios. *Manuales de La CEPAL*.
28. Roy, K., & Marellapudi, V. R. (2023). *Deep Learning in Waste Segregation: Building and Benchmarking a Custom CNN Model*. <https://doi.org/10.13140/RG.2.2.15910.91208>
29. Russell, S., & Norvig, P. (2020). *Artificial Intelligence: A Modern Approach (4th Edition)*. Pearson. https://books.google.com.mx/books/about/Artificial_Intelligence.html?id=koFptAEACAAJ&redir_esc=y
30. Sayem, F. R., Islam, Md. S. Bin, Naznine, M., Nashbat, M., Hasan-Zia, M., Kunju, A. K. A., Khandakar, A., Ashraf, A., Majid, M. E., Kashem, S. B. A., & Chowdhury, M. E. H. (2025).

Enhancing waste sorting and recycling efficiency: robust deep learning-based approach for classification and detection. *Neural Computing and Applications*, 37(6), 4567–4583. <https://doi.org/10.1007/s00521-024-10855-2>

31. Sekar, S. (2018). *Waste Classification data*. <https://www.kaggle.com/datasets/techsash/waste-classification-data>
32. SEMARNAT. (2020, May 15). *Diagnóstico Básico para la Gestión Integral de los Residuos | Instituto Nacional de Ecología y Cambio Climático | Gobierno | gob.mx*. <https://www.gob.mx/inecc/acciones-y-programas/diagnostico-basico-para-la-gestion-integral-de-los-residuos-2020>
33. Shi, C., Tan, C., Wang, T., & Wang, L. (2021). A Waste Classification Method Based on a Multilayer Hybrid Convolution Neural Network. *Applied Sciences* 2021, Vol. 11, Page 8572, 11(18), 8572. <https://doi.org/10.3390/APP11188572>
34. Statista. (2024, September 12). *Blog GR | GR*. <https://www.grmx.com.mx/blog-gr>
35. Thung, G. (2017). *garythung/trashnet: Dataset of images of trash; Torch-based CNN for garbage image classification*. <https://github.com/garythung/trashnet?tab=readme-ov-file>
36. Torralba, A., Isola, P., & Freeman, W. T. (2024). *Foundations of Computer Vision*. MIT Press. <https://mitpress.mit.edu/9780262048972/foundations-of-computer-vision/>
37. UNEP. (2019). *Perspectivas del medio ambiente mundial, GEO-6: planeta saludable, gente saludable*. <https://doi.org/10.1017/9781108627146>
38. Xia, W., Jiang, Y., Chen, X., & Zhao, R. (2022a). Application of machine learning algorithms in municipal solid waste management: A mini review. *Waste Management and Research*, 40(6), 609–624.

https://doi.org/10.1177/0734242X211033716/ASSET/IMAGES/LARGE/10.1177_0734242X211033716-FIG4.JPEG

39. Xia, W., Jiang, Y., Chen, X., & Zhao, R. (2022b). Application of machine learning algorithms in municipal solid waste management: A mini review. *Waste Management and Research*, 40(6), 609–624. https://doi.org/10.1177/0734242X211033716/ASSET/149B0D33-91AB-4DB9-8ED7-91B0F1DC7B15/ASSETS/IMAGES/LARGE/10.1177_0734242X211033716-FIG4.JPG
40. Xiaohoua. (2023). *Garbage_HuaWei*. <https://www.kaggle.com/datasets/xiaohoua/garbage-huawei>
41. Yáñez, J. M. S., & Benavides, L. M. (2024). Gestión de residuos sólidos y la inteligencia artificial en el contexto mexicano. *Ciencia Nicolaita*, 90. <https://doi.org/10.35830/CN.VI90.722>
42. Yang, Z., Xia, Z., Yang, G., & Lv, Y. (2022). A Garbage Classification Method Based on a Small Convolution Neural Network. *Sustainability*, 14(22). <https://doi.org/10.3390/su142214735>
43. Zhang, Q., Yang, Q., Zhang, X., Bao, Q., Su, J., & Liu, X. (2021). Waste image classification based on transfer learning and convolutional neural network. *Waste Management*, 135, 150–157. <https://doi.org/https://doi.org/10.1016/j.wasman.2021.08.038>
44. Zhang, Q., Zhang, X., Mu, X., Wang, Z., Tian, R., Wang, X., & Liu, X. (2021). Recyclable waste image recognition based on deep learning. *Resources, Conservation and Recycling*, 171, 105636. <https://doi.org/https://doi.org/10.1016/j.resconrec.2021.105636>
45. Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, 57(4), 99. <https://doi.org/10.1007/s10462-024-10721-6>

IX. Anexos

Anexo 1. Código para la detección de imágenes duplicadas

```
import os
import sys
import argparse
import hashlib
import shutil

def compute_md5(file_path, chunk_size=8192):
    md5 = hashlib.md5()
    with open(file_path, 'rb') as f:
        while chunk := f.read(chunk_size):
            md5.update(chunk)
    return md5.hexdigest()

def copy_uniques(src_dir, dest_dir):
    seen_hashes = set()
    os.makedirs(dest_dir, exist_ok=True)

    for root, _, files in os.walk(src_dir):
        for fname in files:
            src_path = os.path.join(root, fname)
            if not fname.lower().endswith(('.png', '.jpg', '.jpeg', '.gif', '.bmp', '.tiff')):
                continue

            try:
                file_hash = compute_md5(src_path)
            except Exception as e:
                print(f"[!] Saltar {src_path}: {e}", file=sys.stderr)
                continue

            if file_hash not in seen_hashes:
                seen_hashes.add(file_hash)
                dest_path = os.path.join(dest_dir, fname)
                if os.path.exists(dest_path):
                    base, ext = os.path.splitext(fname)
                    counter = 1
                    while True:
                        new_name = f"{base}_{counter}{ext}"
                        new_path = os.path.join(dest_dir, new_name)
                        if not os.path.exists(new_path):
                            dest_path = new_path
                            break
                    counter += 1
                shutil.copy2(src_path, dest_path)
                print(f"[+] Kept: {src_path}")
            else:
                print(f"[-] Duplicado: {src_path}")

def parse_args():
    parser = argparse.ArgumentParser(
        description="Copia unica de imágenes de SRC to DST mediante la deduplicación de hash.")
    parser.add_argument("src", help="Directorio de origen para buscar imágenes")
    parser.add_argument("dst", help="Directorio de destino para imágenes únicas")
    return parser.parse_args()

if __name__ == "__main__":
    args = parse_args()
    if not os.path.isdir(args.src):
        print(f"Directorio de recursos '{args.src}' no existe.", file=sys.stderr)
        sys.exit(1)
    copy_uniques(args.src, args.dst)
    print("Listo.")
```

Anexo 2. Script de conversión de formato de color de imágenes a RGB de RSU inorgánicos

```
import os
from PIL import Image
ruta_origen = "DATASET_RENOMBRADO/inorganico"
ruta_destino = "dataset_rgb/inorganico"
for archivo in os.listdir(ruta_origen):
    ruta_imagen = os.path.join(ruta_origen, archivo)
    imagen = Image.open(ruta_imagen)
    imagen_rgb = imagen.convert("RGB")
    nombre_imagen, extension = os.path.splitext(archivo)
    ruta_imagen_destino = os.path.join(ruta_destino, f"{nombre_imagen}{extension}")
    imagen_rgb.save(ruta_imagen_destino)
```

Anexo 3. Script de conversión de formato de color de imágenes a RGB de RSU orgánicos

```
import os
from PIL import Image
ruta_origen = "DATASET_RENOMBRADO/organico"
ruta_destino = "dataset_rgb/organico"
for archivo in os.listdir(ruta_origen):
    ruta_imagen = os.path.join(ruta_origen, archivo)
    imagen = Image.open(ruta_imagen)
    imagen_rgb = imagen.convert("RGB")
    nombre_imagen, extension = os.path.splitext(archivo)
    ruta_imagen_destino = os.path.join(ruta_destino, f"{nombre_imagen}{extension}")
    imagen_rgb.save(ruta_imagen_destino)
```

Anexo 4. División del conjunto de datos en entrenamiento y validación

```
train, validation = train_test_split(
    trash_inf_train,
    random_state=42,
    test_size=0.15,
    stratify=trash_inf_train['classification']
)
```

Anexo 5. División del *dataset* en entrenamiento y validación

```
from sklearn.model_selection import train_test_split

train, validation = train_test_split(
    trash_inf_train,
    random_state=42,
    test_size=0.15,
    stratify=trash_inf_train['classification']
)

print(train.shape)
print(validation.shape)
```

(2730, 2)
(482, 2)

Anexo 6. Configuración del aumento de datos en línea

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
trainAug = ImageDataGenerator(
    rescale=1./255,
    rotation_range=5,
    zoom_range=0.10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    shear_range=0.5,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode="nearest"
)
```

Anexo 7. Código de R3sNet

```
1 from tensorflow.keras.models import Model
2 from tensorflow.keras.applications.resnet50 import preprocess_input
3 from keras_flops import get_flops
4
5 def conv_batchnorm_relu(x, filters, kernel_size, strides, kernel_regularizer=None):
6     x = Conv2D(filters=filters,
7               kernel_size=kernel_size,
8               strides=strides,
9               kernel_regularizer=kernel_regularizer,
10              padding='same')(x)
11     x = BatchNormalization()(x)
12     x = ReLU()(x)
13     return x
14
15 def identity_block(tensor, filters, strides):
16     if strides > 1:
17         x = conv_batchnorm_relu(tensor, filters=filters, kernel_size=3, strides=2)
18         x = conv_batchnorm_relu(x, filters=filters, kernel_size=3, strides=1)
19         shortcut = Conv2D(filters=filters, kernel_size=1, strides=2, padding='same')(tensor)
20
21         x = Add()([x, shortcut])
22
23     else:
24         x = conv_batchnorm_relu(tensor, filters=filters, kernel_size=3, strides=1)
25         x = conv_batchnorm_relu(x, filters=filters, kernel_size=3, strides=1)
26         x = Add()([x, tensor])
27     x = ReLU()(x)
28     return x
29
30 def resnet_block(x, filters, reps, strides):
31     for _ in range(reps):
32         x = identity_block(x, filters=filters, strides=strides)
33         if strides > 1:
34             strides -= 1
35
36     return x
37
38 input = Input(shape=(224, 224, 3))
39
40 x = conv_batchnorm_relu(input, filters=16, kernel_size=7, strides=1, kernel_regularizer=12(1e-4)) #capa
41 x = MaxPool2D(pool_size=3, strides=1)(x)
42 x = resnet_block(x, filters=16, reps=2, strides=1)#capa
43 x = resnet_block(x, filters=32, reps=2, strides=2)#capa
44 x = resnet_block(x, filters=64, reps=1, strides=2)#capa
45 x = resnet_block(x, filters=128, reps=1, strides=2) #capa
46
47 x = GlobalAvgPool2D()(x)
48
49 output = Dense(1, activation='sigmoid')(x) #capa
50 model = Model(input, output)
```

Anexo 8. Productos académicos

1. Artículo de Revista Indizada en Scopus y Web of Science (JCR), R3sNet: Optimized Residual Neural Network Architecture for the Classification of Urban Solid Waste via Images, (Castro-Bello, Romero-Juárez, Fuentes-Pacheco, Morales-Morales, et al., 2025).
2. Artículo en revista CONAHCYT, Dataset de clasificación de Residuos Sólidos Urbanos para Redes Neuronales Convolucionales (Castro-Bello et al., 2025).
3. Artículo en extenso en congreso Nacional, Proposal for a Logistics Regression Model for the Classification of Urban Solid Wastes with Image Processing en el 1er Congreso Internacional de Tecnología y Ciencia Aplicada, desarrollado en el Tecnológico Nacional de México/Centro Nacional de Investigación y Desarrollo Tecnológico, TecNM/CENIDET, del 15 al 17 de noviembre de 2023.