



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

**Deep Learning para la clasificación múltiple de residuos sólidos
domésticos**

**TESIS PARA OBTENER EL TÍTULO PROFESIONAL DE:
Ingeniero de Sistemas**

AUTOR (ES):

Vilca Quispe, William Frank (orcid.org/0000-0003-0360-1533)

ASESOR:

Dr. Necochea Chamorro, Jorge Isaac (orcid.org/0000-0002-3290-8975)

LÍNEA DE INVESTIGACIÓN:

Sistema de Información y Comunicaciones

LÍNEA DE RESPONSABILIDAD SOCIAL UNIVERSITARIA:

Desarrollo económico, empleo y emprendimiento

LIMA – PERÚ

2023

DEDICATORIA

A mi querida familia, ustedes han sido mi apoyo incondicional en cada paso de esta travesía. Sus palabras de aliento, sus abrazos reconfortantes y su constante creencia en mí me han dado la fuerza y la confianza para seguir adelante.

A mis padres, su dedicación y sacrificio han sido el cimiento sobre el cual he construido mis logros. Su amor inquebrantable, su ejemplo de perseverancia y su constante motivación me han inspirado a ser la mejor versión de mí mismo.

AGRADECIMIENTO

Quiero agradecer sinceramente al Dr. Jorge Isaac Necochea Chamorro por su invaluable guía y apoyo en este proyecto. Su experiencia y orientación han sido fundamentales para mi éxito, y estoy sinceramente agradecido por su dedicación y compromiso.

A mi familia, quiero expresar mi profundo agradecimiento por su constante apoyo en este proyecto. Su amor y aliento han sido una fuente de motivación inagotable, y estoy realmente agradecido por tenerlos a mi lado. Su confianza en mí y su presencia en mi vida son un regalo invaluable que aprecio enormemente.

Índice de contenidos

I. INTRODUCCIÓN	10
II. MARCO TEÓRICO.....	13
III. METODOLOGÍA.....	20
3.1. Tipo y diseño de investigación:	20
3.1.1. Tipo de investigación:	20
3.1.2. Diseño de investigación:	20
3.2. Variables y operacionalización:.....	21
3.3. Población, muestra y unidad de análisis:	23
3.3.1. Población:	23
3.3.2. Muestra:.....	23
3.3.3. Unidad de análisis:.....	24
3.4. Técnicas e instrumentos de recolección de datos.....	24
3.5. Procedimiento	24
3.6. Método de análisis de datos:.....	31
3.7. Aspectos éticos:.....	34
IV. RESULTADOS.....	35
4.1. Análisis Descriptivo	36
4.1.1. Creación, configuración y entrenamiento de la red neuronal.....	36
4.1.2. Análisis de la precisión, sensibilidad y F1 - Score	39
4.2. Análisis Inferencial	42
4.2.1. Prueba de bondad	42
4.2.2. Prueba U de Mann Whitney	44
V. DISCUSIÓN	49
VI. CONCLUSIONES	55
VI. RECOMENDACIONES	56

REFERENCIAS BIBLIOGRÁFICAS:	58
ANEXOS	64

Índice de tablas

Tabla 1: Detalle de las capas utilizadas en un modelo CNN para un UAV	17
Tabla 2: Ejemplo de positivos y negativos	22
Tabla 3: Técnicas e Instrumentos por cada indicador	24
Tabla 4: Transformaciones aplicadas.....	26
Tabla 5: Distribución del total de fotografías recopiladas.	26
Tabla 6: Detalle de las capas del modelo CNN	28
Tabla 7: Detalle de las capas del modelo propuesto.....	29
Tabla 8: Análisis descriptivo de la Precisión.....	39
Tabla 9: Análisis descriptivo del Recall	40
Tabla 10: Análisis descriptivo del F1-Score.....	41
Tabla 11: Prueba de normalidad - Precisión	43
Tabla 12: Prueba de normalidad - Recall	43
Tabla 13: Prueba de normalidad - F1-Score	44
Tabla 14: Prueba U de Mann Whitney – precisión	45
Tabla 15: Análisis de resultados por clases para precisión.....	45
Tabla 16: Prueba U de Mann Whitney – recall.....	46
Tabla 17: Análisis de resultados por clases para recall.....	47
Tabla 18: Prueba U de Mann Whitney - F1-Score.....	47
Tabla 19: Análisis de resultados por clases para F1-Score	48
Tabla 20: Resultados de las Hipótesis	49
Tabla 21: Comparación de resultados de exactitud	51
Tabla 22: Comparación de resultados de precisión	52
Tabla 23: Comparación de resultados de recall	53
Tabla 24: Comparación de resultados de F1-Score	54

Índice de figuras

Figura 1: Arquitectura del modelo MLH-CNN	16
Figura 2: Arquitectura CNN de un clasificador de equipos e-waste	16
Figura 3: Arquitectura CNN	27
Figura 4: Arquitectura del modelo propuesto.....	29
Figura 5: Diagrama de flujo	31
Figura 6: Diagrama de flujo para elegir un test estadístico inferencial.	32
Figura 7: Transformaciones aplicadas al dataset	35
Figura 8: Resultados de los diez ensayos practicados al modelo propuesto	36
Figura 9: Gráfico de loss y accuracy en el entrenamiento y validación	37
Figura 10: Gráfico de loss y accuracy en el entrenamiento y validación del modelo mejorado	38
Figura 11: Matriz de confusión del modelo mejorado.....	38

RESUMEN

La urbanización mundial está acelerando la producción de residuos sólidos, poniendo en peligro la salud humana como el medio ambiente. Teniendo un rápido desarrollo de la tecnología de aprendizaje profundo, se propone una variedad de modelos de red neuronal, sin embargo estos modelos existentes siguen presentando problemas en la exactitud, precisión, recall y F1 Score. Por ello, el objetivo de este proyecto es lograr una mejora de clasificación en comparación con el modelo CNN, de esta manera se propone un método de clasificación basado en una red neuronal que tiene como base CNN. Este logra obtener mejores resultados al agregar capas, un algoritmo regularizador y el uso de optuna. Por último, la presente investigación demuestra que el modelo propuesto logra mejores resultados que el modelo CNN, asimismo en comparación con algunos trabajos recientes, logra converger en menor épocas que otros métodos propuestos en el estado del arte, asimismo logra mejores resultados de clasificación en ciertas clases de residuos sólidos domésticos. El método propuesto logró una exactitud del 82% superando los resultados obtenidos en otras investigaciones que han utilizado métodos como Efficient-B2, RCNN y SVM.

Palabras Clave: Deep Learning, Clasificación de imágenes, Residuos sólidos domiciliarios, Redes neuronales convolucionales

ABSTRACT

Global urbanization is accelerating the production of solid waste, endangering human health as well as the environment. Having a rapid development of deep learning technology, a variety of neural network models are proposed, however these existing models still present problems in accuracy, precision, recall and F1 Score. Therefore, the objective of this project is to achieve an improvement in classification compared to the CNN model, thus a neural network based classification method based on CNN is proposed. It achieves better results by adding layers, a regularizing algorithm and the use of optune. Finally, the present research demonstrates that the proposed model achieves better results than the CNN model, also in comparison with some recent works, it converges in less time than other methods proposed in the state of the art, it also achieves better classification results in certain classes of domestic solid waste. The proposed method achieved an accuracy of 82% surpassing the results obtained in other researches that have used methods such as Efficient-B2, RCNN and SVM.

Keywords: Deep Learning, Image classification, Household solid waste, Convolutional neural networks.

I. INTRODUCCIÓN

En el Perú, se enfrenta un gran problema con la contaminación por residuos sólidos, afectando al medio ambiente y a la población. Cada año, las cifras generadas superan completamente las del año anterior por lo que es habitual ver cifras exorbitantes año tras año. El MINAM (Ministerio del Ambiente) en el 2020 indicó que el país generó 7.9 millones de toneladas de residuos sólidos municipales [1]. Esta cifra alarmante reflejaba una tendencia creciente que se había mantenido a lo largo de los años y se espera que este siga en aumento.

Lima, la capital del Perú no es ajeno a esto, al tener la tercera parte de la población total este se ha convertido en un cúmulo de contaminación por residuos sólidos. El Instituto Nacional de Estadística e Informática (INEI) reportaba que en 25 distritos de Lima Metropolitana se recogen diariamente más de 100 toneladas de residuos sólidos, indican que los distritos que más residuos sólidos recolectan al día son: San Juan de Lurigancho, Ate, Villa María del Triunfo, Ventanilla, entre otros distritos [2]. Por lo tanto, se eligió el AA. HH. Confraternidad del distrito de Mi Perú para la recolección de residuos sólidos domésticos, esta recolección se realizó con la finalidad de elaborar un dataset propio de residuos sólidos domésticos.

La clasificación de residuos sólidos ha sido aplicada en diversos estudios con robots, UAV [35], incluso el IoT, con la finalidad de que se realice una mejor gestión de residuos sólidos. Sin embargo los modelos basados en CNN utilizados, hablando únicamente de clasificación de residuos sólidos, no logran obtener valores altos en todas las clases de residuos sólidos de estudio, en cuanto a la precisión, sensibilidad y F1-score, el autor [15] indicó que aún existen problemas con los modelos existentes en la clasificación de desechos, como la baja exactitud de clasificación o el largo tiempo de ejecución. En los resultados obtenidos en este artículo se pudo evidenciar una comparativa entre diferentes arquitecturas ampliamente utilizadas en la literatura, como lo es AlexNet, Vgg16 y resnet50, en comparación con un modelo de clasificación multicapa, los modelos fueron entrenados con un dataset muy popular denominado TrashNet, los resultados que obtuvo para cada clase fueron muy variantes, en cuanto a la precisión, recall y F1-

Score, en cada una de estas arquitecturas, se observa que para AlexNet en las clases metal y trash se obtuvo una precisión de 0, mientras que el plástico obtuvo un valor de 0.69, otro valor muy variante es el propuesto por ellos “MLH-CNN” llegando a tener valores muy buenos pero obteniendo para el metal 0.83 mientras que para el papel obtuvo un 1.

En los últimos años, los métodos basados en el aprendizaje profundo se desarrollaron rápidamente. Sin embargo, a menudo es extremadamente difícil recopilar suficientes datos para entrenar un modelo con un buen rendimiento [34], por lo que la creación del dataset ha sido un gran aporte para la literatura.

Este desafío implicaba, en términos de dataset, la elección de una distribución óptima que fue la de 80% para entrenamiento, 10% para validación y 10% para pruebas, según el autor [22]. Por ello es importante la definición de los hiperparámetros del algoritmo planteado y realizar las pruebas de rendimiento con técnicas apropiadas, en este proyecto de investigación se buscó lograr un modelo clasificador de residuos sólidos que logre una mayor eficacia en cada una de las clases sin necesidad de hacer uso de otras tecnologías.

Este proyecto tuvo dos enfoques, la primera fue hacer uso de un modelo basado en la arquitectura CNN tradicional sin ningún tipo de cambio y obtener resultados de precisión, sensibilidad y F1-score, el segundo enfoque fue hacer uso de una arquitectura de CNN personalizada con Optuna y la evaluación de la eficacia con las mismas métricas establecidas. Obtenidos los resultados en ambos enfoques se procedió a realizar la comparación entre ambos enfoques los cuales fueron entrenados con el dataset de elaboración propia de residuos sólidos domésticos, con la finalidad de la identificación qué enfoque logró mejores resultados, además de ello se tuvo en cuenta las arquitecturas personalizadas de la literatura que estén basadas en CNN. Lo obtenido mediante el primer enfoque formó parte del pre-test y lo del segundo enfoque fue el post-test. Ambos enfoques fueron entrenados con el conjunto de datos de imágenes recopiladas a diferentes tasas de aprendizaje, optimizadores, épocas, arquitectura, este adoptó la metodología de iniciar con la recolección de imágenes, construcción del dataset, entrenamiento del modelo,

obtención de los resultados de rendimiento del modelo y la posterior comparación, utilizada por el siguiente autor [35] en su investigación.

Los aportes de la investigación fueron la elaboración de un modelo a partir de una arquitectura personalizada basada en CNN, la comparación del modelo propio con la arquitectura CNN sin modificaciones y la elaboración de un dataset de residuos sólidos domésticos con una cantidad de 3000 imágenes.

El motivo de este proyecto fueron los problemas con los modelos existentes de CNN en la clasificación de residuos sólidos como lo es la baja precisión de la clasificación o incluso los largos tiempos que tardan en entrenarse los modelos por la gran cantidad de parámetros que conllevan, además de lo variables que suelen ser los valores obtenidos para cada una de las clases.

Habiendo expuesto ello, el problema abordado en este proyecto fue: ¿En qué medida el modelo de Deep learning de la CNN modificada mejora la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar?

En base a ello, el objetivo general de este proyecto fue determinar en qué medida el modelo de Deep learning de la CNN modificada mejoró la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar. Para lograr este objetivo se tuvo que determinar el porcentaje de precisión, sensibilidad y F1-score que obtiene el modelo en cada una de las clases y para lograr buenos resultados en cada una de las métricas se tuvo en cuenta, la iluminación, el fondo y la resolución de cada una de las imágenes.

En consecuencia, la hipótesis general planteada fue la siguiente: El modelo de Deep learning de la CNN modificada mejoro la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar.

La investigación consta de cuatro capítulos y ha sido estructurado de la siguiente manera, el primer capítulo es la Introducción, el segundo es el marco teórico, el

tercer capítulo es la Metodología, el capítulo cuatro es resultados, el capítulo cinco es la discusión, el sexto es conclusiones y el último es recomendaciones.

II. MARCO TEÓRICO

El Deep learning es una rama del machine learning, si bien es cierto machine learning es mucho más amplio y comprende de muchos métodos y procedimientos, el deep learning se centra exclusivamente en el uso de redes neuronales artificiales con múltiples capas para modelar y resolver problemas complejos [3].

El Deep Learning consiste en utilizar una red neuronal con múltiples capas para procesar datos de entrada y crear modelos predictivos más precisos y complejos que los algoritmos tradicionales de aprendizaje automático. Esto se logra mediante el uso de técnicas avanzadas de procesamiento de datos que permiten a la red neuronal aprender por sí sola sin la necesidad de una programación específica [3].

Una red neuronal convolucional (CNN) es una arquitectura de deep learning que usualmente se utiliza el procesamiento de imágenes y videos, lo que destaca a CNN es que puede detectar características importantes a partir de una imagen de forma automática. Para ello se usan diversos filtros convolucionales que se aplican con el fin de extraer características como lo son los bordes, texturas, patrones. Esta arquitectura suele utilizarse en una variedad de aplicaciones, tales como reconocimiento de objetos, clasificación de imágenes, detección de rostros, análisis de videos y otros más. La aplicación de esta misma en muchos estudios ha superado el rendimiento humano en tareas relacionadas a la clasificación de imágenes [9] pero reconocer que si el modelo es eficiente y cumple con su función se emplea el uso de métricas.

Para evaluar la eficiencia de un modelo de deep learning para la clasificación se suelen usar métricas que calculen el valor exacto que se desea detectar, incluso existen librerías como Scikit-learn para detectar el valor de estas métricas, trabajadas por el siguiente autor [4]. En cuanto a las métricas se tiene que tener en cuenta que si las clases están desequilibradas puede traer consigo la necesidad de

aplicar nuevas métricas, por lo tanto es esencial tener la cantidad de imágenes de cada clase de forma equilibrada.

Las métricas miden el rendimiento del modelo clasificador. La manera en que realiza la predicción es mediante la proporción obtenida de predicciones correctas en relación con el número total de predicciones realizadas.

En base a ello, la manera en que se mide la precisión es midiendo la proporción de predicciones positivas, se suele utilizar debido a su utilidad cuando se trata de problemas de clasificación binaria en los que se requieren minimizar falsos positivos con la finalidad de obtener una precisión mayor a la actual. Esta métrica se enfoca en medir la cantidad de aciertos del modelo al identificar correctamente las imágenes de una clase en específica, en relación al total de imágenes de esa misma categoría en el conjunto de datos tal cual como lo han realizado estos autores [5], [6], [7], [8], [9] y [10].

El recall es importante para evaluar la habilidad del modelo en detectar correctamente las imágenes de una clase en particular como se puede observar en estos artículos [5], [6], [7], [8] y [9].

En el Perú los residuos sólidos se clasifican como residuos sólidos orgánicos, residuos sólidos inorgánicos, residuos sólidos inorgánicos no reciclables y residuos sólidos peligrosos indicados por el Ministerio del Ambiente [11]. Además de ello, en el Perú se promueve la separación de los residuos sólidos en los lugares donde son generados, pueden ser empresas, instituciones, hogares, entre otros, con el objetivo de facilitar la gestión y tratamiento [11]. Por lo tanto en esta investigación se han utilizado los residuos sólidos domésticos inorgánicos que se generan en los hogares del AA. HH. Confraternidad Sector C de Mi Perú, específicamente en el Sector C Tercer Sector con la finalidad de cumplir con este decreto supremo y realizar la clasificación de los residuos sólidos domésticos inorgánicos desde donde son generados.

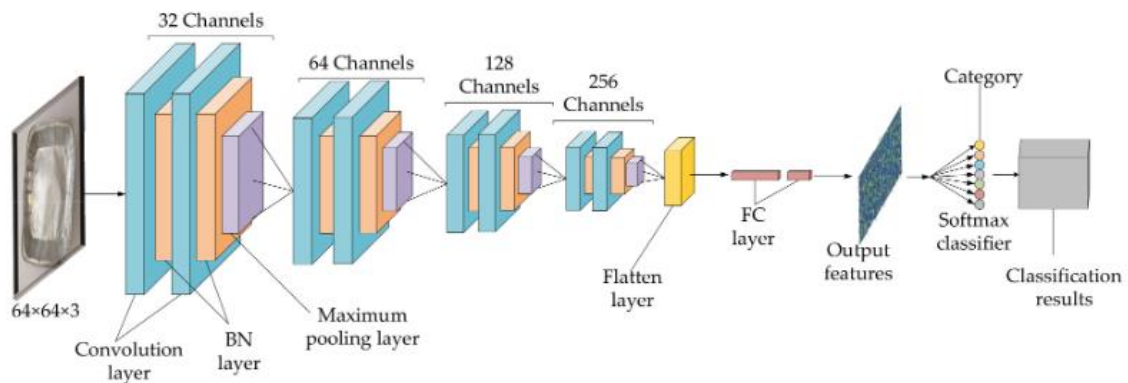
En la literatura se ha identificado la clasificación de residuos sólidos domésticos en diferentes clases, como lo son: Cartón, Vidrio, Metal, Papel, Plástico y Residuos orgánicos [9], [12]. Por ello mismo este proyecto se basará en 5 clases como residuos sólidos, como el Papel, Cartón, Plástico, Metal y Vidrio dentro de estas clases estará especificado los objetos con los que se ha entrenado al modelo para la clasificación.

Revisando en la literatura se ha podido identificar diversos estudios relacionados al deep learning y la eficacia de este mismo relacionado con la clasificación de residuos sólidos, donde se ha podido evidenciar investigaciones que combinan algoritmos de redes neuronales convolucionales con otras arquitecturas, como AlexNetCNN, DBN y optimización optuna como propuesta de modelo de deep learning híbrido [13] o incluso la combinación de red neuronal de convolución (CNN) y memoria a corto plazo Graph (GLS TM) con la finalidad de obtener resultados muy buenos en comparación a otros modelos, tanto en respuesta como en distintas como lo son la precisión, accuracy, recall y F1 [9]. Además se han utilizado modelos pre-entrenados como ResNet-50 pero haciendo un valor agregado al combinarlo con SVM para la clasificación de residuos sólidos [14], otros estudios relacionados a CNN usan este algoritmo haciendo una hibridación multicapa [15] mientras otros autores han utilizado Faster R-CNN para la clasificación de residuos sólidos.[16]. Otros estudios para la misma finalidad de la clasificación de residuos sólidos han trabajado con transfer learning y CNN, haciendo uso de un modelo pre-entrenado denominado DenseNet169 [17]. Sin embargo un estudio indica que las arquitecturas como VGGNet, AlexNet, ResNet y DenseNet demuestran ser exitosos para la clasificación de imágenes con alta precisión, pero estos modelos pre-entrenados suelen consumir demasiados recursos por lo que en su investigación elaboran su propio modelo de clasificación basado en CNN [18]. El siguiente estudio hace uso de la arquitectura CNN [24] y logró un accuracy de 98,33%, no indica la cantidad de épocas necesarias para este estudio sin embargo es un porcentaje bastante cercano a lo que podemos considerar perfecto.

Asimismo existen investigaciones que se centran en la creación de modelos con la finalidad de obtener una mayor eficiencia que la CNN tradicional, como por ejemplo

lo realizado por el autor [15] el cual propone un modelo multicapa denominado MLH-CNN con la finalidad de lograr mejores resultados en comparación con otros modelos de clasificación pero utilizando como base a CNN. A continuación se muestra la arquitectura empleada en su investigación, además de realizar su propia arquitectura, probó modificaciones en cuanto al optimizador, learning rate y el batch_size. En la **figura 1** se muestra la arquitectura empleada para elaborar su modelo clasificador.

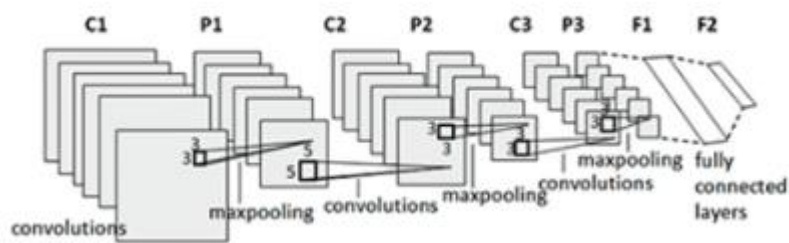
Figura 1: Arquitectura del modelo MLH-CNN



Fuente: [15]

El autor [36], puso a prueba su modelo basado en CNN para la clasificación de equipos electrónicos desechados, utilizando la siguiente arquitectura de CNN, ello se puede apreciar en la **figura 2**.

Figura 2: Arquitectura CNN de un clasificador de equipos e-waste



Fuente: [36]

El autor [35] da una mayor descripción de su modelo propuesto, mostrado en la **tabla 1**, el realizó un modelo para la clasificación de residuos sólidos con un UAV, obtuvo resultados buenos y su metodología consiste en la comparación de dos modelos y el uso del que obtuvo mejores resultados de clasificación, donde su modelo propuesto logró un 94% de accuracy en comparación con su segundo modelo que utiliza el optimizador Adam.

Tabla 1: Detalle de las capas utilizadas en un modelo CNN para un UAV

Sr. No.	Input Image Size	Layer Name	Filter Size	No. of Filter	No. of Parameter	Output Image Size
1	224x224x3	Conv2D	3x3	16	448	224x224x16
2	224x224x16	Maxpool Layer	2x2	-	-	112x112x16
3	112x112x16	Conv2D	3x3	16	2,320	112x112x16
4	112x112x16	Maxpool Layer	2x2	-	-	56x56x16
5	56x56x16	Conv2D	3x3	32	4,640	56x56x32
6	56x56x32	Maxpool Layer	2x2	-	-	28x28x32
7	28x28x32	Conv2D	3x3	32	9,248	28x28x32
8	28x28x32	Maxpool Layer	2x2	-	-	14x14x32
9	14x14x32	Conv2D	3x3	64	18,496	14x14x64
10	14x14x64	Maxpool Layer	2x2	-	-	7x7x64
11	7x7x64	Dropout_Layer(0.4)	-	-	-	7x7x64
12	7x7x64	Flatten Layer	-	-	-	3136
13	3136	Dense Layer_1 (512)	-	-	1,606,144	512
14	512	Dense Layer_2 (1)	-	-	513	1

Fuente: [35]

En una investigación reciente, el autor [12], hace uso de un algoritmo de optimización nuevo denominado Optuna, el cual le facilitó la obtención de los hiperparametros de su modelo híbrido basado en AlexNetCNN.

Por otro lado también otro estudio demuestra la clasificación de residuos sólidos por el nivel de vida utilizando redes neuronales recurrentes (RNN) para una posterior predicción de residuos utilizando DLTSF basado en un LSTM [19]. Asimismo podemos observar el uso de EfficientNet-B0 [20] o la creación de

algoritmos haciendo uso de CSPdarknet-53, CSPNet y Darknet-53 basándose en YOLO V4 obteniendo un valor de recall del 92% y un average IoU del 63.58% [21].

Si hablamos de técnicas para mejorar la precisión del modelo, hablamos de data augmentation e incluso uno de los más populares denominado dropout usados en investigaciones obteniendo buenos resultados de precisión [18].

En cuanto a las métricas de eficacia, es muy común ver en las investigaciones el uso de estas [9], [12], [19], [21], [17], [15] y [18], debido a que es necesaria evaluar la eficacia de los modelos de clasificación e incluso compararlos con los modelos ya existentes [9], [12], [15] y [17]. Las métricas que se ha podido evidenciar en la literatura que son usadas con mayor frecuencia son las matrices de confusión, la exactitud, la precisión, la recuperación y la puntuación F1.

En cuanto a los datasets utilizados podemos destacar que en ciertas investigaciones realizadas se han elaborado un propio dataset de imágenes de residuos sólidos [9] y [22], mientras que otras investigaciones han optado por utilizar datasets ya creados como el uso del dataset de ImageNet [12], [20] y [23] utilizado para la clasificación donde se usó la clase ImageDataGenerator de Keras para el aumento de datos [18], el dataset de Waste Pictures [13], TrashNet [15], [14] y [16], Trash Annotations in Context (TACO) con [21], NWNUTRASH con una cantidad de imágenes de 18911 [17], sin embargo para obtener una precisión buena en cuanto a las pruebas realizadas al modelo, la presente investigación elaborará un propio dataset para entrenar al modelo.

Se ha podido evidenciar que los residuos sólidos obtenidos de los hogares se clasifican en las siguientes clases: Cartón, Vidrio, Metal, Papel, Plástico y Residuos orgánicos [9], [12] y [15]. Ciertos autores no incluyen el metal [19] o cambian la clase de residuos orgánicos por basura [16].

Es importante también identificar los porcentajes de imágenes de entrenamiento, validación y prueba, la literatura demuestra la mejor manera para dividirlos, en diversos estudios se utiliza un porcentaje similar de imágenes de entrenamiento y

de prueba en las investigaciones, en este estudio se utilizó un total de 80% y 20%, de datos de entrenamiento y datos de prueba, respectivamente [9], [12] y [22]. Un estudio realizó una comparación entre los porcentajes de prueba, validación y prueba, con dos grupos de valores distintos, determinando que la división entre un 80% para entrenamiento y el 20% dejándolo para la prueba y validación reducen el porcentaje de la tasa de error, repartiéndose de la siguiente manera 10 % y 10%, respectivamente [22].

En cuanto a los resultados, se identifica que estos son muy variables pero ninguno obteniéndose valores por debajo del 70%, en algunos al usar datasets ya elaborados suelen tener resultados distintos ya que depende mucho de la iluminación y calidad con la que han sido tomadas las imágenes de entrenamiento y con qué imágenes se realizan las pruebas, ello puede variar el porcentaje de accuracy considerablemente, se puede comprobar en estas dos investigaciones donde se obtuvo un total de 87% de accuracy [15], sin embargo en cuanto a otra investigación realizada con imágenes de prueba de ImageNet en las evaluaciones experimentales obtuvo una precisión de 99,5% [12], sin embargo otra investigación usando este mismo dataset pero usando otro modelo obtuvo una precisión entre 74% y 84% [20], mediante el uso de otro dataset como Waste Pictures se obtuvo una precisión del 75% [13], mientras que mediante el uso de TrashNet se logró una precisión de 92,6 % [15], otros con porcentajes cercanos al 91% utilizando un repositorio de GitHub con contenido de imágenes de residuos sólidos [16]. También se ha podido evidenciar el uso de NWN-TRASH obteniendo una exactitud de 82,8 %.

Otras investigaciones con datasets de elaboración propia se obtienen mejores resultados, incluso llegando a superar otros modelos conocidos en el mundo de Deep learning 97.5% [9], otros lograron un accuracy del 95.31% [22].

Separándolos un tanto de la clasificación de residuos sólidos como tal, también se han encontrado estudios que demuestran el uso de deep learning para la clasificación de diversos objetos, como lo es el metal por ejemplo, usando las mismas técnicas de recolección, apoyándose de datasets y usando modelos para

clasificación como lo es R-CNN que es un modelo pre-entrenado basado en redes neuronales convolucionales, posterior a ello se hace la debida evaluación de las métricas de eficacia [6]. En cuanto a comparaciones de arquitecturas un estudio de deep learning demuestra que CNN basada en el método de transfer learning logra una precisión más alta utilizado para la clasificación de residuos sólidos domésticos en instituciones médicas, obviamente la clasificación es en distintas clases, pero el proceso a seguir usado es la misma que las mencionadas anteriormente [25].

Además de ello, un estudio demuestra la importancia de tener un fondo realista, por ello realizan un dataset donde superponen un fondo más realista para los residuos sólidos que tienen fondos uniformes en específico el fondo blanco, el propósito es apoyar en la clasificación de residuos sólidos para entornos urbanos y salvajes [26].

III. METODOLOGÍA

3.1. Tipo y diseño de investigación:

3.1.1. Tipo de investigación:

El tipo de investigación aplicada genera conocimientos que pueden ser utilizados en la solución de problemas prácticos sus resultados son empleados de forma inmediata, a corto o a mediano plazo, para solventar problemas sociales, administrativos, educativos, de salud, entre otros [27]. Por lo tanto, el tipo de investigación que se utilizó para este proyecto fue investigación aplicada.

3.1.2. Diseño de investigación:

El diseño es experimental es un proceso cuya principal característica es verificar cuantitativamente la causalidad de una variable sobre otra. En los diseños experimentales, la variable independiente es aquella que el investigador manipula y/o controla con el propósito de evaluar sus efectos

sobre la variable dependiente [28]. El diseño experimental tiene tres tipos, los cuales son el pre-experimental, cuasi experimental y experimental puro. El pre-experimental tiene grupo o sujetos ya conformados, se puede aplicar un pre-test y/o post-test y las mediciones se realizan en no más de dos tiempos diferentes [28], de esa manera se trabajó con dos grupos experimentales por lo que se realizó un pre-test y un post-test. En base a ello, el diseño de investigación que se utilizó es experimental de tipo pre-experimental.

3.2. Variables y operacionalización:

La variable dependiente (VD) en este presente trabajo fue la “Clasificación **de residuos sólidos domésticos**” y la variable independiente (VI) fue el “**Deep Learning**” ambos pertenecientes a la categoría cuantitativa.

Variable dependiente: Clasificación de residuos sólidos domésticos

Definición Conceptual: Los residuos sólidos, basuras, desechos o desperdicios generados en viviendas y en establecimientos tales como edificios habitacionales, locales comerciales, locales de expendio de alimentos, hoteles, establecimientos educacionales y cárceles [29].

Definición Operacional: Los residuos sólidos domésticos reciclables se clasifican en Papel, Cartón Plástico, Metal y Vidrio [12].

Dimensiones:

Papel, Cartón, Plástico, Metal y Vidrio

Indicadores: A cada una de las dimensiones se le aplicaron las métricas de precisión, sensibilidad y F1-score utilizadas también en la literatura en temas relacionados por los siguientes autores [5], [6], [7], [8] y [9], para evaluar la eficacia de clasificación del modelo en cada una de las clases de residuos sólidos domésticos.

La fórmula de la métrica de precisión [6] es la siguiente:

$$PRE = \frac{TP}{TP + FP}$$

La fórmula de la métrica de sensibilidad [6] es la siguiente:

$$REC = \frac{TP}{TP + FN}$$

Y la fórmula de la métrica de F1-Score [4] es la siguiente:

$$F1 = 2 * \frac{PRE * REC}{PRE + REC}$$

Donde PRE es igual al valor de la precisión y REC es el valor obtenido por la sensibilidad o también llamado Recall.

En la **tabla 2**, se muestra lo que significa TP, TN, FP y FN, y cómo se originan.

Tabla 2: Ejemplo de positivos y negativos

Tipo	P (Positivo)	N(Negativo)
T(Verdadero)	Verdadero Positivo (TP)	Verdadero Negativo (TN)
F(Falso)	Falso Positivo (FP)	Falso Negativo (FN)

Fuente: [6]

Variable independiente: Deep Learning

Definición Conceptual: El aprendizaje profundo es una parte del campo del aprendizaje automático, que se centra en el desarrollo de redes neuronales con tres o más capas. Su objetivo principal es utilizar grandes volúmenes de datos para "aprender" y mejorar su desempeño [3].

Todo lo expresado se puede ver en detalle en la matriz de operacionalización presentada en el **anexo 1**.

Indicador:

La exactitud se mide del modelo en general, por lo que ha sido un buen indicador para saber la exactitud con la que el modelo basado en CNN modificado hace una correcta clasificación de residuos sólidos.

La fórmula de la métrica de exactitud [6] es la siguiente:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN}$$

3.3. Población, muestra y unidad de análisis:**3.3.1. Población:**

La población fue de 4890 Imágenes de Residuos Sólidos Domésticos recolectados de las viviendas del AA. HH. Confraternidad Sector C - Mi Perú. Para las imágenes recolectadas se aplicaron los siguientes criterios de inclusión y exclusión.

Criterios de inclusión: Se incluyeron las imágenes nítidas y con una buena iluminación

Criterios de exclusión: Se excluyeron las Imágenes de residuos sólidos que no pertenecían al grupo de residuos sólidos domésticos.

3.3.2. Muestra:

El tamaño de la muestra se determinó en función de los objetivos de la investigación, y en este caso, todas las imágenes de la población participaron del estudio, donde se tuvo dos grupos uno conformado por imágenes de un dataset existente ampliamente utilizado en la literatura denominado TrashNet [15], el mismo que contiene 2390 imágenes de las mismas clases de residuos sólidos domésticos tratados en esta investigación y el otro grupo conformado por un dataset que elabore, el cual tiene 2500 imágenes, el 80% fue para el entrenamiento del modelo, mientras que el 10% para la validación y para las pruebas se usaron un 10% del total.

Por lo tanto, no se utilizó la selección de una muestra representativa ni se aplicó una técnica de muestreo.

3.3.3. Unidad de análisis:

La unidad de análisis fueron las fotografías de residuos sólidos domésticos de las viviendas del AA. HH. Confraternidad Sector C - Mi Perú.

3.4. Técnicas e instrumentos de recolección de datos

La recolección de datos se realizó aplicando la técnica de observación directa. En base a ello, se realizaron fichas de registro de observaciones. La ficha de registro se emplea cuando el investigador desea obtener información detallada sobre un objetivo específico con el propósito de medir, analizar o evaluar. Esta herramienta permite recolectar datos tanto de aspectos externos como internos de las personas, como sus actividades, emociones y situaciones en las que se encuentran [28].

Tabla 3: Técnicas e Instrumentos por cada indicador

Indicadores	Técnica	Instrumento
Precisión	Observación Directa	Ficha de registro
Recall	Observación Directa	Ficha de registro
F1-Score	Observación Directa	Ficha de registro

Elaboración Propia

El instrumento fue confiable y válido debido al amplio uso de estos mismos en la literatura [4] y [6] que han sido utilizados con la finalidad de evaluar el rendimiento del modelo.

3.5. Procedimiento

Recolección de residuos:

Este estudio analizó los residuos sólidos domésticos, por lo que para la creación del dataset de imágenes se hizo una recolección de residuos sólidos. La recolección se dio en el sector C para la obtención de las 3000 imágenes de los residuos sólidos que las personas desecharon, estos no necesitarán un consentimiento informado debido a que no se recolecta información sensible, ni se usarán sus nombres para ningún fin, esta recolección se basa en obtener residuos sólidos domésticos que corresponden a la clase vidrio, papel, cartón, plástico o metal, con la única finalidad de tomar fotografías. Esta recolección tendrá una duración de dos semanas.

Toma de Fotografías:

Obtenidos los residuos sólidos se tomaron las fotografías, donde se utilizó la cámara de un teléfono Android, específicamente el dispositivo móvil Samsung Galaxy S20+ con una resolución de 64 megapíxeles (MP). El fondo donde se tomó las fotografías fue de color blanco y se tuvo la iluminación artificial que se suele tener en los hogares, como los focos de luz que se tienen en habitaciones. Además, se realizó un pre-procesamiento de la imagen para quitar la sombra que pueda sobreponerse en la imagen, esto fue realizado con Photoshop para tener una imagen limpia y que este mismo cumpla con los criterios establecidos.

Data Augmentation:

Para aumentar el número de imágenes se utilizó la técnica de Data Augmentation para lograr obtener una mayor cantidad de imágenes aplicando rotación, cambios de tamaño, etc, hasta lograr obtener las 3000 imágenes que conformó el dataset elaborado, este fue subido a drive para que pueda ser utilizado en otras investigaciones.

Este aumento de datos se dio aplicando 6 transformaciones en la imagen original. A continuación, se muestra detalladamente estas transformaciones y sus valores definidos en la **tabla 4**.

Tabla 4: Transformaciones aplicadas

Transformación	Valor
Rotación	$\pm 20^\circ$
Desplazamiento Horizontal	± 0.2
Desplazamiento Vertical	± 0.2
Deformación	± 0.2
Zoom	± 0.2
Volteo Horizontal	True

Elaboración Propia

Habiendo llegado a las 3000 imágenes, se realizó una distribución del dataset elaborado tal como se muestra en la **Tabla 5**.

Tabla 5: Distribución del total de fotografías recopiladas.

Distribución	Porcentaje del Total de Imágenes
Entrenamiento	80%
Validación	10%
Prueba	10%

Elaboración Propia

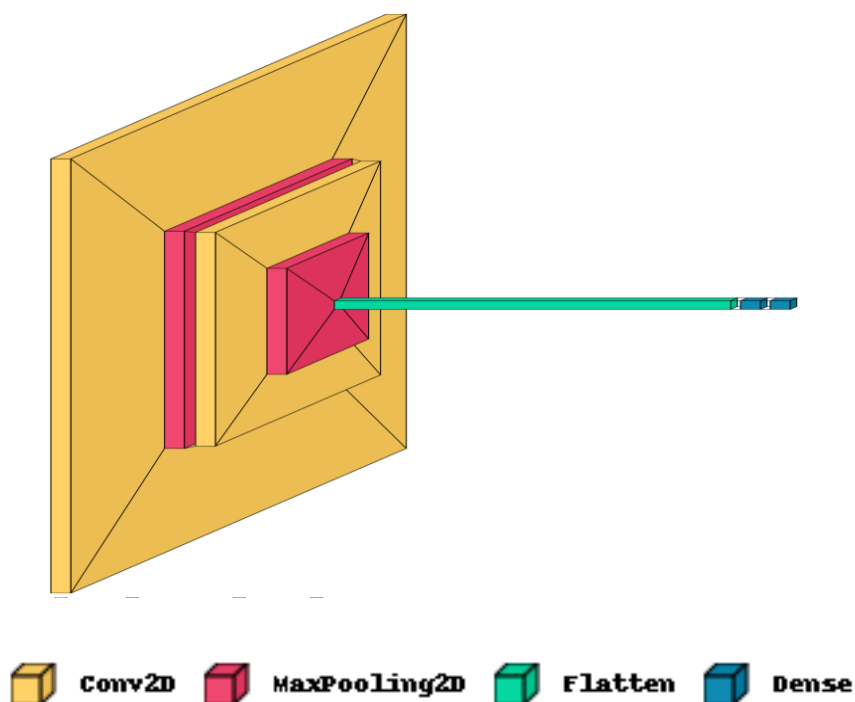
Arquitectura CNN:

El modelo propuesto que se utilizó estuvo basado en CNN pero fue modificado con la finalidad de lograr mejores resultados, se establecieron los

hiper-parámetros, dentro de los cuales se definieron el número de capas, número de épocas, la función de activación, la resolución de la imagen de la capa de entrada, el lote de entrenamiento, número de épocas y la capa de salida. Habiendo construido el modelo fue entrenado con el dataset elaborado, esto formó parte del Post-test.

Sin embargo fue necesario que se realice un pre-test para evaluar los resultados que se logran con el modelo propuesto y el modelo base sin ningún tipo de cambio, habiendo entrenado ambos modelos con ambos dataset que forman parte de mis muestras, una vez entrenado este modelo de CNN con mi dataset y con el dataset de TrashNet será registrado en mi ficha de registro del pre-test, donde los valores de las métricas obtenidas serán registrados para cada clase. La arquitectura del modelo utilizado para el pre-test, fue basado en CNN, este mismo se puede ver en la **Figura 3**, para más detalle acerca del modelo se puede observar en la **Tabla 6**, donde se muestra el total de parámetros y las capas utilizadas.

Figura 3: Arquitectura CNN



Elaboración Propia

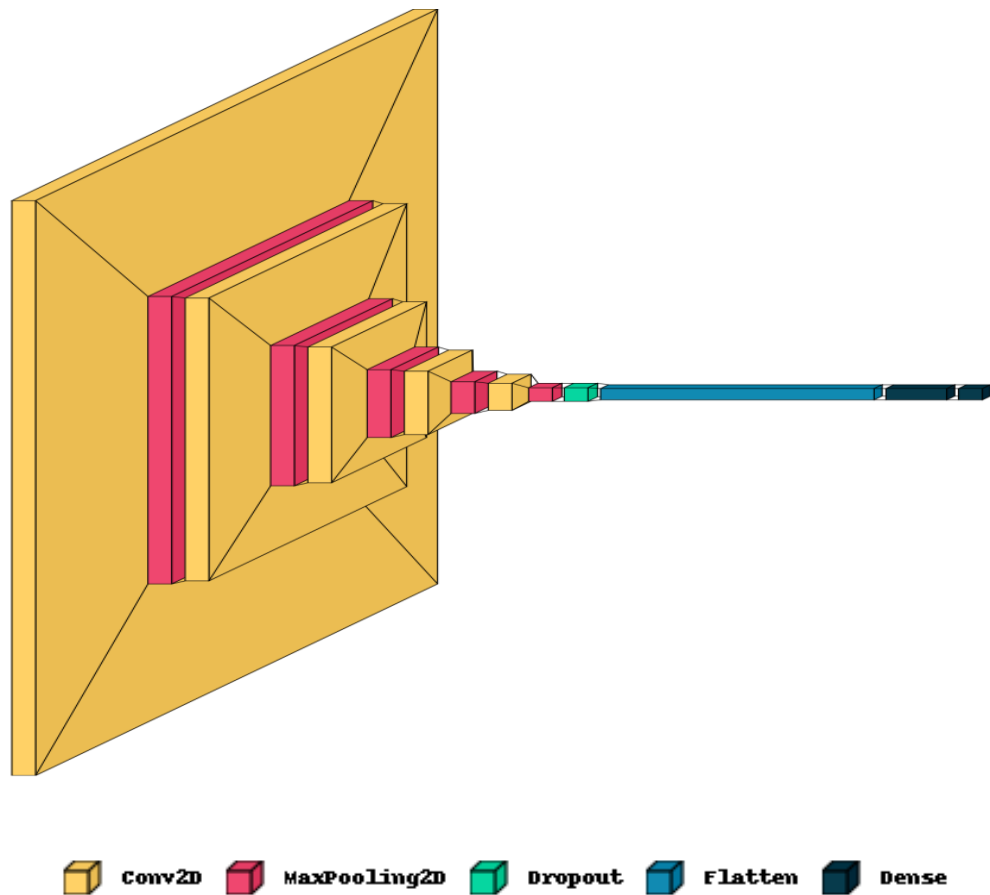
Tabla 6: Detalle de las capas del modelo CNN

Layer (type)	Output Shape	Param #
conv2d_3 (Conv)	(None, 254, 254, 32)	896
max_pooling2d_3 (MaxPooling 2D)	(None, 127, 127, 32)	0
conv2d_4 (Conv2D)	(None, 125, 125, 64)	18,496
max_pooling2d_4 (MaxPooling2D)	(None, 62, 62, 64)	0
flatten_1 (Flatten)	(None, 246016)	0
dense_3 (Dense)	(None, 64)	15,745,088
dense_4 (Dense)	(None, 5)	325
	Total Parámetros:	15,764,805

Elaboración Propia

Para el post-test, se realizó un nuevo modelo basado en mi arquitectura con modificaciones previamente expresadas, donde hice la medición después del entrenamiento para cada una de las clases, para posteriormente registrar los valores obtenidos en la ficha de registro de post-test. La arquitectura que siguió el modelo, fue el que se muestra en la **Figura 4** y el resumen del modelo se muestra en la **Tabla 7** teniendo menos cantidad de parámetros que el modelo base, aplicando técnicas como lo es el aumento de datos y realizando modificaciones en los hiperparámetros del modelo propuesto, validando la función de activación utilizada, el tamaño del lote o batch_size a utilizar, el optimizador, número de capas, cantidad de filtros

Figura 4: Arquitectura del modelo propuesto



Elaboración Propia

Tabla 7: Detalle de las capas del modelo propuesto

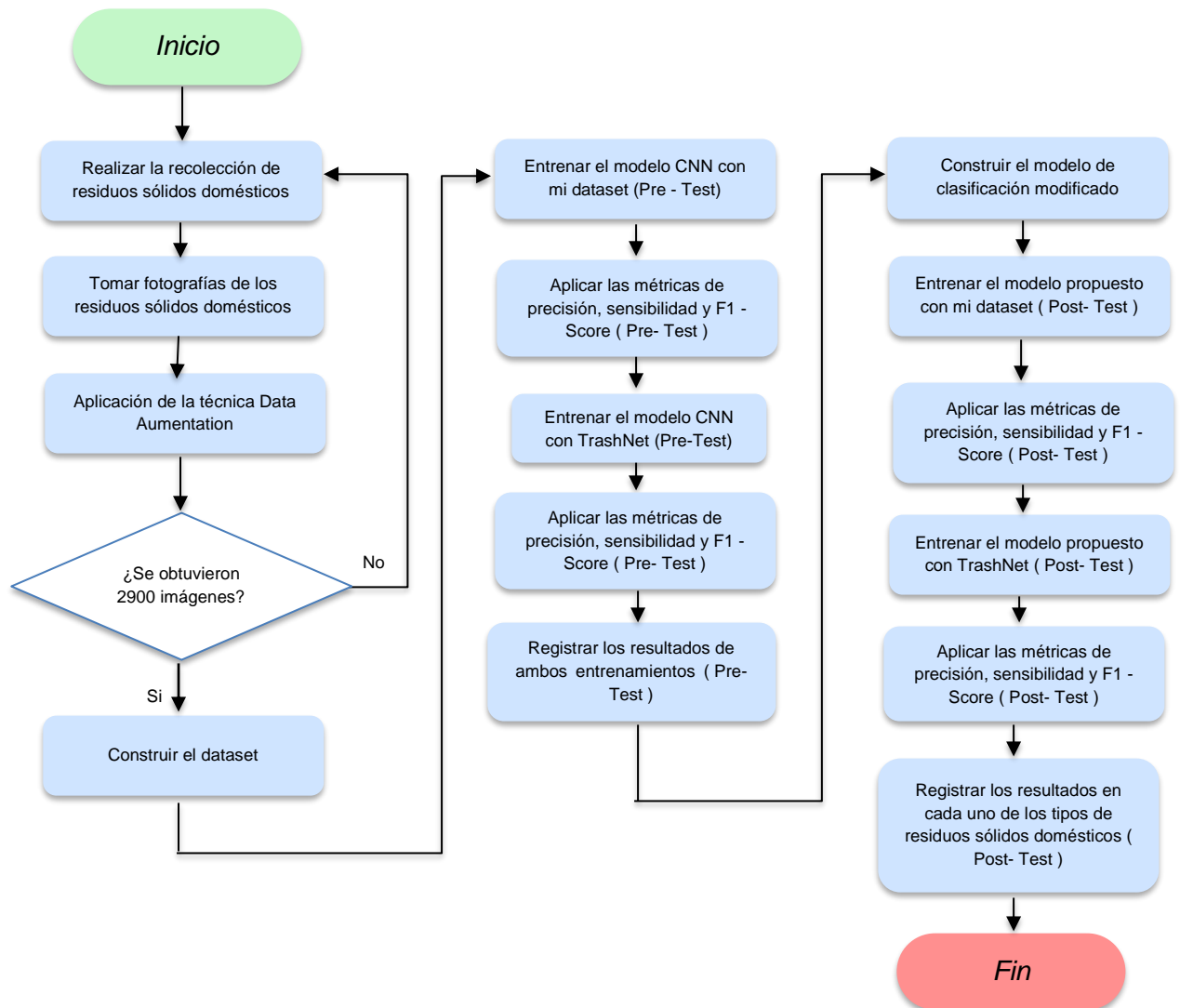
Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 254, 254, 38)	1064
max_pooling2d (MaxPooling2D)	(None, 127, 127, 38)	0
conv2d_1 (Conv2D)	(None, 125, 125, 54)	18522
max_pooling2d_1 (MaxPooling2D)	(None, 62, 62, 54)	0
conv2d_2 (Conv2D)	(None, 60, 60, 19)	9253
max_pooling2d_2 (MaxPooling2D)	(None, 30, 30, 19)	0

conv2d_3 (Conv2D)	(None, 28, 28, 36)	6192
max_pooling2d_3 (MaxPooling2D)	(None, 14, 14, 36)	0
conv2d_4 (Conv2D)	(None, 12, 12, 30)	9750
max_pooling2d_4 (MaxPooling2D)	(None, 6, 6, 30)	0
dropout (Dropout)	(None, 6, 6, 30)	0
flatten (Flatten)	(None, 1080)	0
dense (Dense)	(None, 1019)	1101539
dense_1 (Dense)	(None, 5)	5100
	Total Parámetros:	1,151,420

Habiendo obtenido resultados buenos para la clasificación de los tipos de residuos sólidos domésticos, se pudo dar por hecho de que el modelo cumple el objetivo propuesto en este estudio.

Resumiendo todo lo explicado en este punto, se elaboró un diagrama de flujo que refleja lo expuesto, ello se muestra en la **Figura 5**.

Figura 5: Diagrama de flujo



Elaboración Propia

3.6. Método de análisis de datos:

El presente trabajo utilizó el enfoque analítico descriptivo e inferencial. Para el análisis descriptivo se usó SPSS debido a que esta es una herramienta que analiza datos y realiza un análisis estadístico.

Para el análisis inferencial se llevó a cabo un análisis de normalidad en los datos recopilados para cada indicador, con el objetivo de determinar si siguen una distribución normal o no.

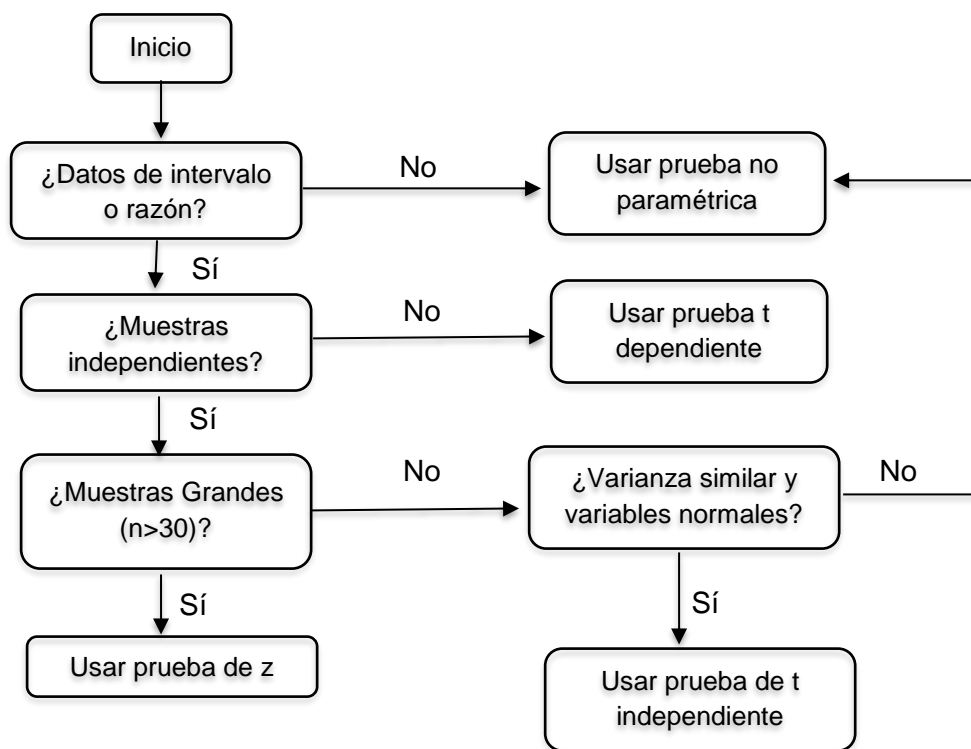
Para determinar la normalidad de los datos, se estableció el siguiente criterio:

Si:

$p < 0.05$ se considera que los datos no siguen una distribución normal

$p \Rightarrow 0.05$ se consideró que los datos siguen una distribución normal

Figura 6: Diagrama de flujo para elegir un test estadístico inferencial.



Fuente: [31].

Si los datos presentan una distribución normal, se aplicará la prueba paramétrica para evaluar las hipótesis y determinar si se confirman o se refutan. En el caso de que los datos no sigan una distribución normal, se emplea la prueba no paramétrica para el contraste de hipótesis y tomar decisiones de aceptación o rechazo. En la **Figura 6** se muestran los caminos a seguir según el contexto de la investigación,

al tener muestras grandes y si es paramétrica se utilizará la prueba de T-Student y si es no paramétrica se usará la prueba de Mann – Whitney U.

Si consideramos que " $\alpha=0.05=5\%$ ", cuando " $p<\alpha$ ", se rechaza la hipótesis alternativa y se acepta la hipótesis nula. En caso contrario, se acepta la hipótesis alternativa.

Hipótesis General:

Hipótesis Alternativa (Ha): El modelo de Deep learning de la CNN modificada mejora la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar

Hipótesis Nula (H0): El modelo de Deep learning de la CNN modificada no logra una mejora la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar

Hipótesis Especifica 1:

Hipótesis Alternativa (Ha): El valor de precisión de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar.

Hipótesis Nula (H0): El valor de precisión de cada una de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada no logra una mejora en comparación con el modelo CNN estándar

Hipótesis Especifica 2:

Hipótesis Alternativa (Ha): El valor de sensibilidad de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar

Hipótesis Nula (H0): El valor de sensibilidad de cada una de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada no logra una mejora en comparación con el modelo CNN estándar

Hipótesis Específica 3:

Hipótesis Alternativa (Ha):

El valor de F1 - Score de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar

Hipótesis Nula (H0): El valor de F1 - Score de cada una de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada no logra una mejora en comparación con el modelo CNN estándar

Asimismo el procesamiento de datos se realizará haciendo uso de SPSS.

3.7. Aspectos éticos:

La presente investigación se rige bajo los principios de originalidad y disposiciones del código de ética del 2021 de la Universidad César Vallejo en su artículo 9° de la política anti plagio.

Donde se indica que se considera plagio toda conducta por la cual se hace pasar como propio un trabajo. Por lo tanto en este proyecto de investigación se cita correctamente las fuentes de consulta como finalidad de reconocer los derechos de autor y ceñirse a los estándares de la Universidad César Vallejo. Asimismo, los resultados serán obtenidos a partir del modelo de clasificación elaborado.

En cuanto a la obtención de los residuos sólidos domésticos, se recolecta de las viviendas, sin embargo no se estableció un consentimiento informado debido a que no se manejó información sensible, ni se evaluó ningún tipo de comportamiento de

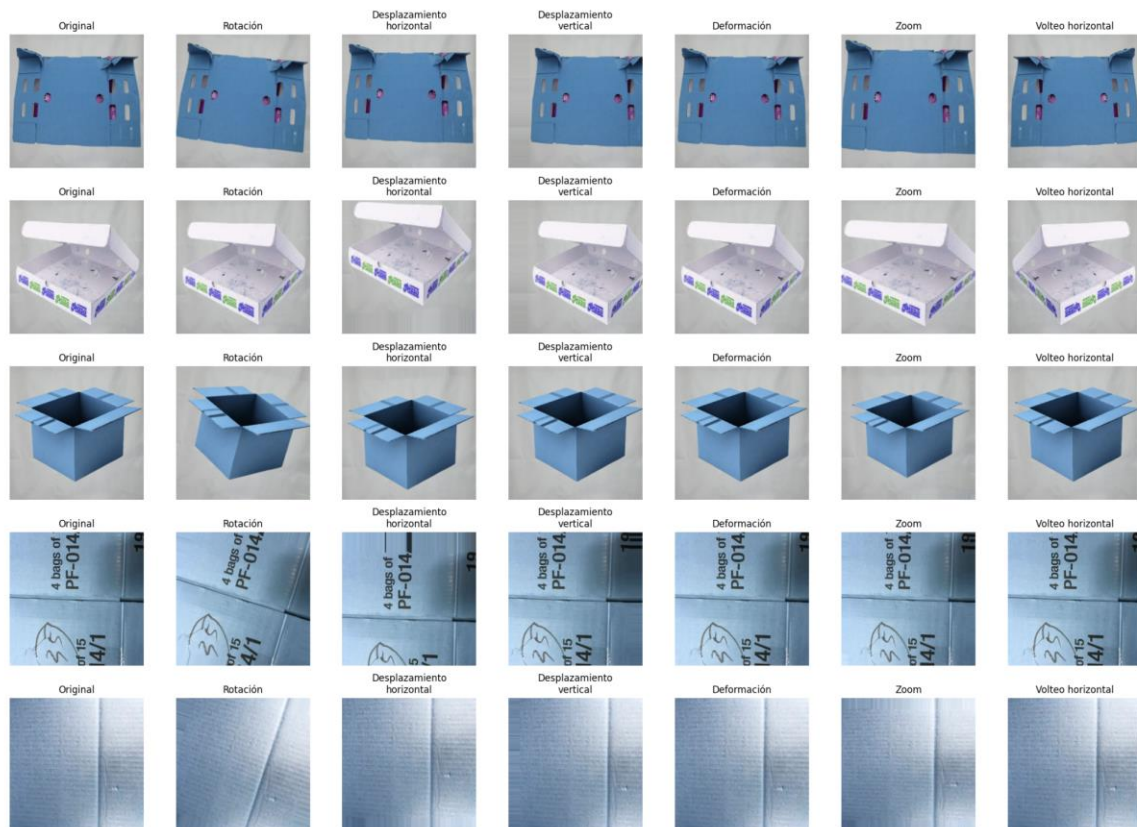
las personas, la única finalidad fue obtener residuos sólidos para posteriormente tomar fotografías con el objetivo de construir el dataset que utilizó el modelo.

IV. RESULTADOS

Transformaciones aplicadas

Obtenidas las imágenes se procedió a aplicar ciertas transformaciones como se observa en mayor detalle y los valores de este mismo en la **Tabla 3**, un ejemplo de las transformaciones aplicadas a las imágenes se muestra en la **Figura 7**. Estas transformaciones fueron aplicadas tomando en cuenta la técnica de aumento de datos donde se buscó ampliar la cantidad con pequeños cambios en las imágenes de manera aleatoria, el cual fue aplicado en cada iteración del proceso de entrenamiento del modelo.

Figura 7: Transformaciones aplicadas al dataset



Elaboración Propia

4.1. Análisis Descriptivo

4.1.1. Creación, configuración y entrenamiento de la red neuronal

Finalizado con mi dataset se procedió a realizar la creación del modelo de clasificación el cual fue puesto a prueba haciendo el uso de Optuna, la cual es una librería donde se pueden obtener los mejores hiperparámetros convenientes para el modelo diseñado, este lo realiza a través de ensayos y del ensayo que obtuvo un mejor resultado se obtiene los hiperparámetros y se establece en el modelo a entrenar, este modelo es preparado previamente para que los valores de los hiperparámetros tomen valores dentro de un rango, para que de esta forma se realizan los ensayos, el modelo definido se puede ver en el **anexo 5**. Los resultados que se obtuvieron en 10 ensayos haciendo uso de optuna se pueden observar en la **Figura 8**:

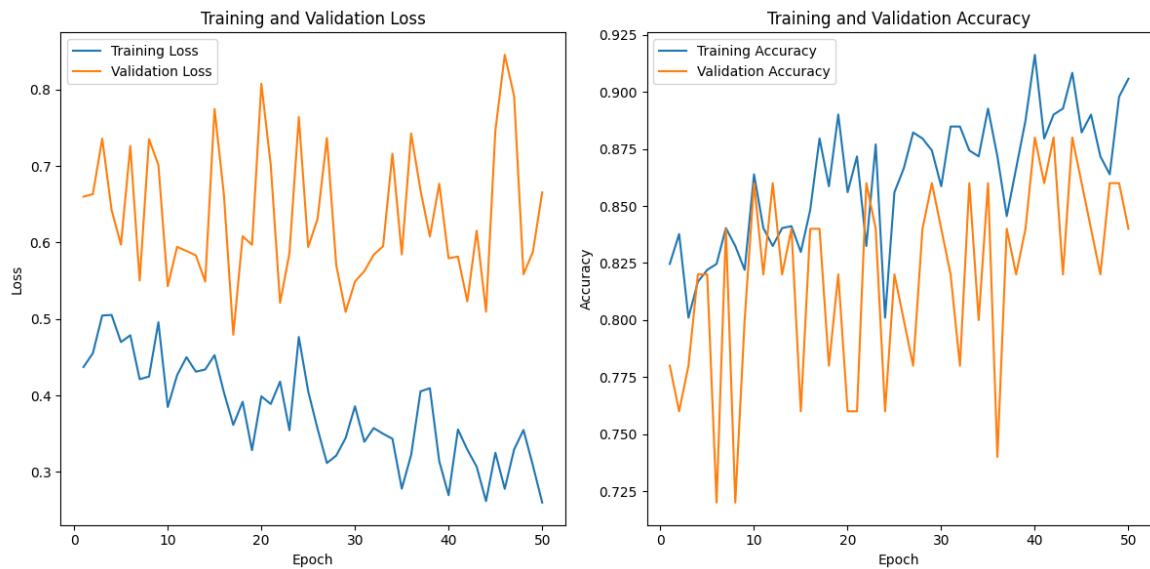
Figura 8: Resultados de los diez ensayos practicados al modelo propuesto

```
Trial 0 finished with value: 0.680000071525574 and parameters: {'conv1_units': 26, 'conv2_units': 49, 'conv3_units': 16, 'conv4_units': 59, 'conv5_units': 63, 'dropout': 0.3747843036489893,
Trial 1 finished with value: 0.779999713897705 and parameters: {'conv1_units': 59, 'conv2_units': 26, 'conv3_units': 19, 'conv4_units': 29, 'conv5_units': 64, 'dropout': 0.3882215931083668,
Trial 2 finished with value: 0.660000262260437 and parameters: {'conv1_units': 64, 'conv2_units': 21, 'conv3_units': 31, 'conv4_units': 40, 'conv5_units': 29, 'dropout': 0.3958219778227354,
Trial 3 finished with value: 0.720000286102295 and parameters: {'conv1_units': 34, 'conv2_units': 16, 'conv3_units': 24, 'conv4_units': 24, 'conv5_units': 63, 'dropout': 0.279411471979359,
Trial 4 finished with value: 0.69999980979071 and parameters: {'conv1_units': 37, 'conv2_units': 53, 'conv3_units': 38, 'conv4_units': 56, 'conv5_units': 59, 'dropout': 0.21508325221626434,
Trial 5 finished with value: 0.680000071525574 and parameters: {'conv1_units': 43, 'conv2_units': 63, 'conv3_units': 52, 'conv4_units': 50, 'conv5_units': 17, 'dropout': 0.4489454663543839,
Trial 6 finished with value: 0.720000286102295 and parameters: {'conv1_units': 41, 'conv2_units': 35, 'conv3_units': 31, 'conv4_units': 64, 'conv5_units': 34, 'dropout': 0.2455533944248521,
Trial 7 finished with value: 0.2800000298023224 and parameters: {'conv1_units': 30, 'conv2_units': 21, 'conv3_units': 46, 'conv4_units': 36, 'conv5_units': 60, 'dropout': 0.4277845979477804,
Trial 8 finished with value: 0.839999737739563 and parameters: {'conv1_units': 38, 'conv2_units': 54, 'conv3_units': 19, 'conv4_units': 36, 'conv5_units': 30, 'dropout': 0.4290712723669397,
Trial 9 finished with value: 0.2800000298023224 and parameters: {'conv1_units': 45, 'conv2_units': 24, 'conv3_units': 17, 'conv4_units': 62, 'conv5_units': 40, 'dropout': 0.1691250446009212
```

Elaboración propia

Por lo que el modelo se estableció con los hiper parámetros del ensayo n° 8 al ser el que mejores resultados obtuvo, mostrado en el **anexo 5**.

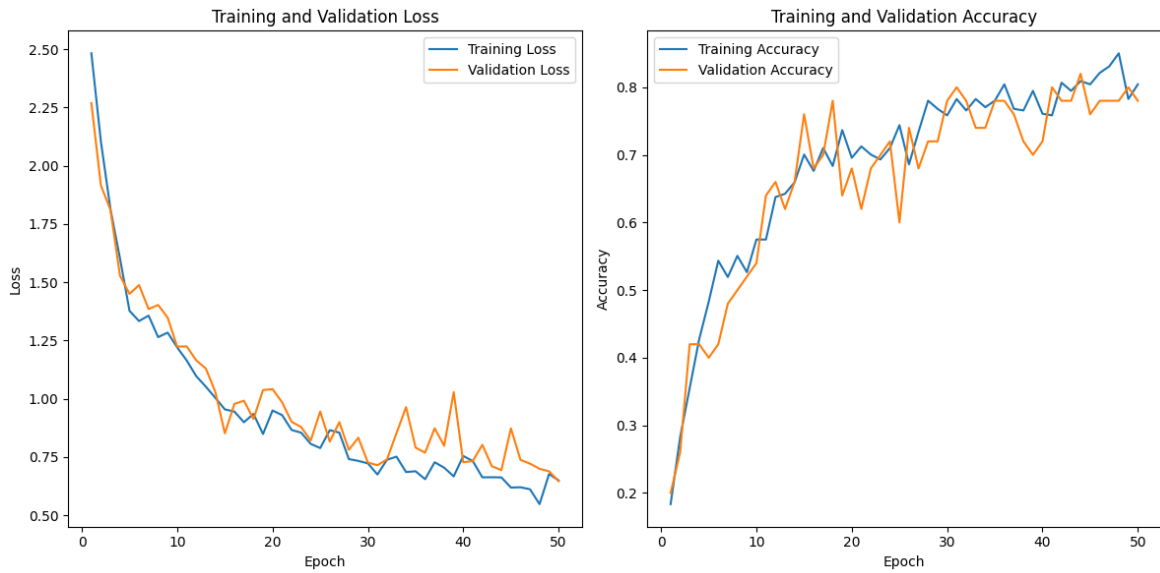
Figura 9: Gráfico de loss y accuracy en el entrenamiento y validación



Elaboración propia

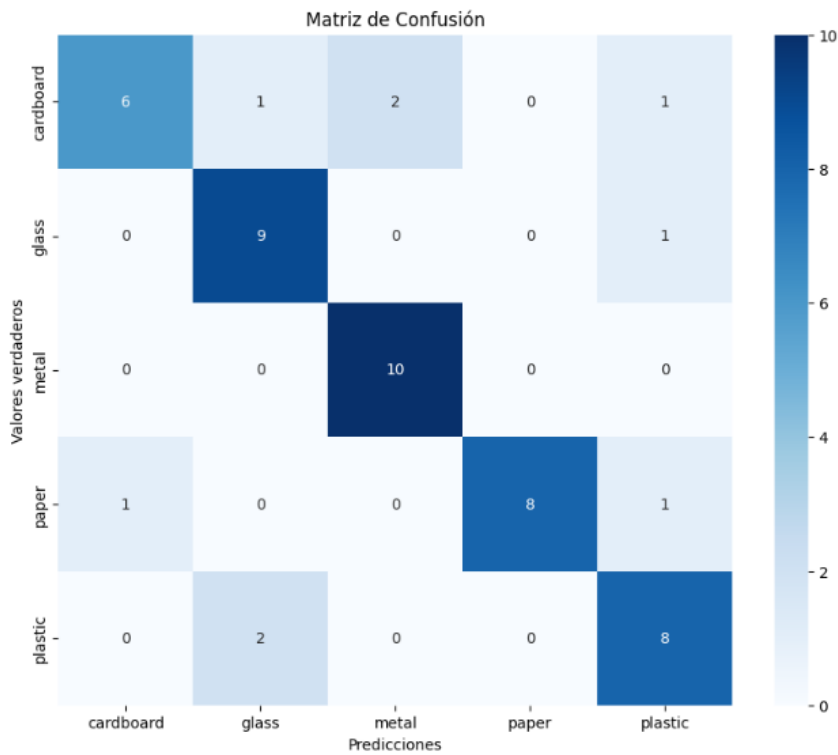
A pesar de ello, hubo un problema de overfitting o sobreajuste como se observa en la **Figura 9**, por lo tanto se procedió a utilizar un algoritmo de regularización L2, bastante utilizada en deep learning, este fue aplicado con la finalidad de que generalice de una mejor manera y no tome el conjunto de entrenamiento como una verdad absoluta, este lo realiza mediante la penalización de predicciones durante el entrenamiento, lo mencionado es observable en la sección del **Anexo 5**, aplicado ello se procedió a realizar un nuevo entrenamiento donde se logró una mejora clara y que los resultados logren estabilidad, ello se puede observar en la **Figura 10**, asimismo se procedió a generar la matriz de confusión obteniendo resultados favorables y con una tendencia bastante baja a que el modelo pueda confundir su predicción por otra clase, mostrado en la **Figura 11**.

Figura 10: Gráfico de loss y accuracy en el entrenamiento y validación del modelo mejorado



Elaboración propia

Figura 11: Matriz de confusión del modelo mejorado



Elaboración propia

Además de ello se vio una mejora en los resultados obtenidos para cada métrica evaluada en esta investigación, dichos resultados se pueden ver a detalle en **anexos 3**.

4.1.2. Análisis de la precisión, sensibilidad y F1 - Score

A continuación se muestran los resultados obtenidos del conjunto de validación que tiene un total de 50 imágenes (10% del total), que ha sido utilizado para obtener los resultados de las métricas de precisión, recall y f1-score, estos están basados en los resultados obtenidos de las fichas de observación elaboradas (**ver anexo 3**) donde se ha utilizado el modelo base, el modelo propuesto y dos datasets, uno elaborado por mí y el otro es uno muy utilizado en el estado del arte, denominado TrashNet, de tal manera que los resultados obtenidos por cada clase han sido sometidos a un análisis descriptivo.

- **Indicador 1: Precisión**

En la **Tabla 8** se muestra el análisis descriptivo de los resultados obtenidos en las cinco clases de residuos sólidos domésticos con la finalidad de determinar la precisión.

Tabla 8: Análisis descriptivo de la Precisión

Análisis estadístico descriptivo - Precisión				
	Valor Mínimo	Valor Máximo	Media	Desviación Estándar
Pre - Test	,35	,89	,6620	,17912
Post - Test	,60	1,00	,7670	,11528

Elaboración propia

De este análisis descriptivo se observó que en el pre-test se llega a obtener un valor mínimo del 35% y como máximo 89%, sin embargo en el post-test se evidenció una notable mejora llegando a tener como un valor mínimo de 60% y un valor máximo del 100% que es lo máximo que puede lograr un modelo. En cuanto a la media el post-test logró una mejora de un 10% aprox. tomando en cuenta los valores obtenidos en cada clase de residuos sólidos domésticos tratados. Por último, en la desviación estándar se notó una mejora igualmente significativa donde podemos decir que incluso los valores obtenidos mediante mi modelo planteado son menos dispersos por cada clase por un 0.06 aprox. de mejora demostrando que el modelo propuesto logró la mejora requerida en cuanto a precisión.

- **Indicador 2: Recall**

En la **Tabla 9** se muestra el análisis descriptivo de los resultados obtenidos en las cinco clases de residuos sólidos domésticos con la finalidad de determinar la sensibilidad.

Tabla 9: Análisis descriptivo del Recall

Análisis estadístico descriptivo - Recall				
	Valor Mínimo	Valor Máximo	Media	Desviación Estándar
Pre - Test	,36	1,00	,6510	,20245
Post - Test	,57	1,00	,7480	,14513

Elaboración propia

En la **Tabla 9** se observó que en el pre-test se llega a obtener un valor mínimo del 36% mientras que en el post-test un valor mínimo del 57% logrando una mejora significativa, mientras que los valores máximos se mantienen con un 100%

demostrando que mi modelo propuesto no logra resultados menores al del pre-test. En cuanto a la media el post-test logró una mejora de un 9% aprox. tomando en cuenta los valores obtenidos en cada clase de residuos sólidos domésticos tratados. Por último, en la desviación estándar se notó una mejora igualmente significativa donde se pudo concluir que los valores obtenidos mediante mi modelo planteado son menos dispersos por cada clase por un 0.06 aprox. de mejora demostrando que el modelo propuesto logró la mejora requerida en cuanto al recall.

- **Indicador 3: F1-Score**

En la **Tabla 10** se muestra el análisis descriptivo de los resultados obtenidos en las cinco clases de residuos sólidos domésticos con la finalidad de determinar el F1-Score.

Tabla 10: Análisis descriptivo del F1-Score

Análisis estadístico descriptivo - F1 - Score				
	Valor Mínimo	Valor Máximo	Media	Desviación Estándar
Pre - Test	,37	,87	,6450	,17096
Post - Test	,60	,91	,7500	,10165

Elaboración propia

En la **Tabla 10** se observó que en el pre-test se llega a obtener un valor mínimo del 37% y como máximo 87%, sin embargo en el post-test se evidenció una notable mejora llegando a tener como un valor mínimo de 60% y un valor máximo del 91% que es lo máximo que puede lograr un modelo. En cuanto a la media el post-test logra una mejora de un 11% aprox. tomando en cuenta los valores obtenidos en cada clase de residuos sólidos domésticos tratados. Por último, en la desviación estándar se notó una mejora igualmente significativa donde podemos decir que

incluso los valores obtenidos mediante mi modelo planteado son menos dispersos por cada clase por un 0.07 aprox. de mejora demostrando que el modelo propuesto logró la mejora requerida en cuanto a precisión.

4.2. Análisis Inferencial

De acuerdo a las hipótesis planteadas se buscó realizar un análisis para determinar si se acepta la hipótesis nula o la hipótesis alternativa, teniendo en cuenta ello, primero se aplicó una prueba de normalidad para cada indicador.

4.2.1. Prueba de bondad

Esta prueba de normalidad se realizó con la finalidad de obtener la distribución de los datos, por lo tanto se planteó como hipótesis de normalidad lo siguiente:

Ho = Los datos siguen una distribución normal

Ha = Los datos no siguen una distribución normal

Nivel de significancia:

Confianza: 95%

Significancia (ALFA): 5%

Decisión:

- Si $p \leq 0.05$, se rechaza la Ho y se acepta la Ha, por lo tanto se utiliza una prueba no paramétrica.
- Si $p > 0.05$, se acepta la Ho y se rechaza la Ha, por lo tanto se utiliza una prueba paramétrica

Sabiendo que son 10 los resultados obtenidos en el pretest y otros 10 de los resultados obtenidos en el post-test para cada indicador, se debe destacar que estos son resultados representativos de las 2500 imágenes utilizadas para esta

investigación que han sido obtenidos de los resultados en cada entrenamiento tanto como al modelo base como al modelo propuesto, por lo tanto se procedió a utilizar la prueba de shapiro - wilk debido a que la muestra es menor igual a 50.

Tabla 11: Prueba de normalidad - Precisión

Prueba de normalidad							
	Muestras	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
precision_pre	TrashNet	,233	5	,200*	,916	5	,507
	Propio	,193	5	,200*	,975	5	,908
precision_post	TrashNet	,207	5	,200*	,951	5	,746
	Propio	,204	5	,200*	,920	5	,531

Elaboración propia

Tabla 12: Prueba de normalidad - Recall

Pruebas de normalidad							
	Muestras	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
recall_pre	TrashNet	,237	5	,200*	,914	5	,494
	Propio	,345	5	,053	,863	5	,238
recall_post	TrashNet	,292	5	,188	,863	5	,241
	Propio	,246	5	,200*	,956	5	,777

Elaboración propia

Tabla 13: Prueba de normalidad - F1-Score

Pruebas de normalidad							
	Muestras	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Estadístico	gl	Sig.	Estadístico	gl	Sig.
f1Score_pre	TrashNet	,207	5	,200*	,906	5	,444
	Propio	,210	5	,200*	,931	5	,605
f1Score_post	TrashNet	,261	5	,200*	,823	5	,124
	Propio	,202	5	,200*	,941	5	,675

Elaboración propia

Habiendo realizado la prueba de normalidad se obtuvo que $p > 0.05$ en los 3 indicadores, mostrados en detalle en la **Tabla 11**, **Tabla 12** y **Tabla 13**, al seguir los datos una distribución normal se acepta la H_0 y se rechaza la H_a , de esta manera se demuestra que siguen una distribución normal y tendrían que ser evaluados con una prueba paramétrica, sin embargo la cantidad de muestras es muy pequeña para aplicar una prueba paramétrica, es por ello que la prueba más apropiada para muestras pequeñas es una prueba no paramétrica, debido a que este está centrado para muestras pequeñas menores a 20 y una prueba paramétrica se vuelve menos eficaz con muestras pequeñas [37], se ha tomado esta decisión debido a que para optar por una prueba paramétrica se debe cumplir con los supuestos que este conlleva, al incumplir con algún supuesto de este mismo, se opta automáticamente por una prueba no paramétrica [37], por lo tanto la prueba con la que se realizó el estadístico inferencial es la prueba no paramétrica, denominada u de mann whitney para dos muestras independientes.

4.2.2. Prueba U de Mann Whitney

Se debe tener en cuenta que la prueba U de Mann Whitney ha sido realizado en SPSS y el valor que ha sido considerado para rechazar o aceptar la hipótesis nula

fue la significación exacta o unilateral, debido a que las hipótesis están planteadas en ese contexto.

Tabla 14: Prueba U de Mann Whitney – precisión

Estadísticos de prueba ^a		
	precision_pre	precision_post
U de Mann-Whitney	1,500	2,500
W de Wilcoxon	16,500	17,500
Z	-2,305	-2,095
Sig. asintótica(bilateral)	,021	,036
Significación exacta [2*(sig. unilateral)]	,016 ^b	,032 ^b

Elaboración propia

El resultado que brindo la prueba para el indicador de precisión en los dos grupos es 0,032, el valor que nos da la significación unilateral, en consecuencia se tiene las suficientes pruebas para rechazar la hipótesis nula y aceptar la hipótesis alternativa, debido a que $p < 0,05$, por lo tanto podemos afirmar que en cuanto a precisión, las medias de los resultados obtenidos con mi modelo propuesto logro una mejora respecto a los resultados obtenidos con el modelo base en ambos grupos (datasets).

Tabla 15: Análisis de resultados por clases para precisión

Dataset	Método	Cartón	Vidrio	Metal	Papel	Plástico
TrashNet	Modelo Base	71%	48%	35%	67%	45%
Propio	Modelo base	83%	77%	67%	80%	89%
TrashNet	Modelo Propuesto	83%	72%	60%	67%	68%
Propio	Modelo Propuesto	86%	75%	83%	100%	73%

Elaboración propia

En cuanto a la precisión obtenido en cada una de las clases, visto en la **Tabla 15**, el modelo propuesto logró una mejora del 3% en cardboard, una disminución del 2% en vidrio, un aumento del 16% en metal, un 20% en papel y una disminución del 16% en plástico en comparación del modelo base, hablando de mi dataset, en cuanto a mi dataset. Por otro lado, hablando del dataset de TrashNet se vieron mejoras igualmente donde mi modelo propuesto logró una mejora del 12% en cartón, 24% en vidrio, 25% en metal, 0% en papel y 23% en plástico, demostrando que el modelo propuesto logra una mejora notable en comparación con el modelo base, los resultados se pueden en detalle en **anexos 3**.

Tabla 16: Prueba U de Mann Whitney – recall

Estadísticos de prueba ^a		
	recall_pre	recall_post
U de Mann-Whitney	3,000	2,500
W de Wilcoxon	18,000	17,500
Z	-2,009	-2,095
Sig. asintótica(bilateral)	,045	,036
Significación exacta [2*(sig. unilateral)]	,056^b	,032^b

Elaboración propia

El resultado que brinda la prueba para el indicador de recall en los dos grupos es 0,032, el valor que nos da la significación unilateral, en consecuencia se tiene las suficientes pruebas para rechazar la hipótesis nula y aceptar la hipótesis alternativa, debido a que $p < 0,05$, por lo tanto podemos afirmar que en cuanto al recall, las medias de los resultados obtenidos con mi modelo propuesto logra una mejora respecto a los resultados obtenidos con el modelo base en ambos grupos (datasets) a diferencia del modelo base que incluso toma un valor por encima del 0.05.

Tabla 17: Análisis de resultados por clases para recall

Dataset	Método	Cartón	Vidrio	Metal	Papel	Plástico
TrashNet	Modelo Base	60%	36%	40%	57%	68%
Propio	Modelo base	50%	100%	80%	80%	80%
TrashNet	Modelo Propuesto	62%	77%	60%	82%	57%
Propio	Modelo Propuesto	60%	90%	100%	80%	80%

Elaboración propia

En cuanto al recall obtenido en cada una de las clases, visto en la **Tabla 17**, el modelo propuesto logró una mejora del 10% en cartón, 20% en metal, una disminución en vidrio del 10% y se mantuvo los resultados en papel y plásticos, en cuanto a mi dataset. Por otro lado, en TrashNet se evidenció mejoras con mi modelo propuesto en un 2% en cartón, 41% en vidrio, 20% en metal, 25% en papel y una disminución de un 11% en plástico, demostrando que se logra una mejora notable en comparación con el modelo base, los resultados se pueden en detalle en **anexos 3**.

Tabla 18: Prueba U de Mann Whitney - F1-Score

Estadísticos de prueba ^a		
	f1Score_pre	f1Score_post
U de Mann-Whitney	1,500	2,500
W de Wilcoxon	16,500	17,500
Z	-2,305	-2,102
Sig. asintótica(bilateral)	,021	,036
Significación exacta [2*(sig. unilateral)]	,016^b	,032^b

Elaboración propia

El resultado que brindó la prueba para el indicador de F1-Score en nuestras dos muestras es 0,032, el valor que nos da la significación unilateral, en consecuencia

se tiene las suficientes pruebas para rechazar la hipótesis nula y aceptar la hipótesis alternativa, debido a que $p < 0,05$, por lo tanto se pudo afirmar que en cuanto a precisión, las medias de los resultados obtenidos con mi modelo propuesto logro una mejora respecto a los resultados obtenidos con el modelo base en ambos grupos (datasets).

Tabla 19: Análisis de resultados por clases para F1-Score

Dataset	Método	Cartón	Vidrio	Metal	Papel	Plástico
TrashNet	Modelo Base	65%	41%	37%	62%	54%
Propio	Modelo base	62%	87%	73%	80%	84%
TrashNet	Modelo Propuesto	71%	74%	60%	74%	62%
Propio	Modelo Propuesto	71%	82%	91%	89%	76%

Elaboración propia

Como último indicador, hablando del F1-Score obtenido en cada una de las clases, visto en la **Tabla 19**, se logró una mejora del 9% en cartón, 18% en metal, 9% en papel y una disminución del 5% en vidrio y 8 en plastic, en cuanto a mi dataset. Por otro lado, en TrashNet, se evidenció mejoras con mi modelo propuesto con un 6% en cartón, 33% en vidrio, 23% en metal, 12% en papel y 8% en plástico, demostrando que el modelo propuesto logro una mejora notable en comparación con el modelo base, los resultados se pueden en detalle en **anexos 3**.

En la siguiente tabla, se puede observar los resultados de las hipótesis planteadas en resumen de la investigación, demostrando que se ha cumplido los objetivos propuestos y las hipótesis formuladas.

Tabla 20: Resultados de las Hipótesis

Hipótesis Planteada	Resultados de Hipótesis
<p>HE1: El valor de precisión de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar.</p>	<p>Aceptado</p>
<p>HE2: El valor de sensibilidad de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar.</p>	<p>Aceptado</p>
<p>HE3: El valor de F1 - Score de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar.</p>	<p>Aceptado</p>

V. DISCUSIÓN

En la presente investigación se analizaron dos métodos de clasificación de residuos sólidos domésticos, tomando en cuenta las clases de cartón, vidrio, metal, papel y plástico. La primera fue la clasificación mediante un algoritmo muy utilizado en el estado del arte denominado CNN, mientras el segundo método es mi modelo propuesto mejorando el modelo CNN aplicando técnicas de optimización y un

algoritmo que reemplaza la forma tradicional de prueba de un modelo clasificador durante el entrenamiento, donde el objetivo fue lograr una mejora en precisión, recall y f1-score en cuanto al modelo clasificador de residuos sólidos domésticos.

Teniendo en cuenta, el desarrollo de la red neuronal y entendiendo que las redes neuronales convolucionales extraen características de las imágenes que se les suministren mediante el entrenamiento, una parte muy importante para garantizar la eficacia del modelo clasificador es lograr un buen conjunto de datos que puedan brindarle todas las características que este necesita para cumplir su objetivo. En la literatura, existe una diversidad de datasets como uno de los trabajados que es TrashNet [15] con más de 2100 imágenes, MJU-Waste [13] con 2475 imágenes o el del autor [35] con 2000 imágenes, los tres basados en residuos sólidos domésticos. Es por ello que la cantidad de imágenes que se consideró para construir el dataset, es de 2500 imágenes logrando estar dentro de los rangos de cantidades de imágenes utilizadas por diversos autores en sus investigaciones.

El modelo propuesto logró un accuracy del 82% mientras que el modelo base logró un 78%, el modelo propuesto converge con 50 épocas, teniendo una mejora de un 4%, ambos trabajando con una cantidad similar de residuos sólidos, en la literatura el autor [38] logró un 84% aplicando Faster RCNN combinándolo con ResNet 50 sin embargo su modelo converge con 120 épocas, más del doble del que se utilizó para el modelo, el autor [17] logró un 82% aplicando CNN y Transfer Learning, sin embargo este modelo converge con una cantidad de 300 épocas, el autor [15] logró un 92,6% con 30 épocas aplicando un modelo multicapa modificado basándose en VGGNet, sin embargo un modelo híbrido multicapa requiere un alto costo computacional mientras que el autor [13] logró un 75% haciendo uso de Efficient-Net-B2 en 20 épocas, asimismo se superó los resultados obtenidos por los autores [39] y [40], demostrando que el modelo propuesto no solo logra mejoras en el modelo base, sino que también ha logrado superar a ciertos modelos propuestos en la literatura ya sea en cuanto a épocas, accuracy o costo computacional, en la **Tabla 21** se puede ver de manera más detallada lo mencionado.

Tabla 21: Comparación de resultados de exactitud

Artículos	Dataset	Método	Accuracy	Épocas	Porcentaje de mejora
[38]	TrashNet	Faster RCNN - ResNet50	84%	120	-2%
[17]	TrashNet	CNN y Transfer Learning	82%	300	0%
[15]	TrashNet	MLH-CNN (Multicapa Híbrido)	92%	30	-10%
[13]	TrashNet	Efficient-B2	75%	20	7%
[39]	TrashNet	Augmented train data - RCNN	68.30%	-	13.7%
[40]	TrashNet	SVM	80%	-	2%
-	TrashNet	Modelo base	51%	30	31%
-	Propio	Modelo base	78%	50	4%
-	TrashNet	Modelo Propuesto	69%	30	13%
-	Propio	Modelo Propuesto	82%	50	-

Elaboración Propia

En cuanto a precisión el modelo propuesto logró una mejora en comparación del modelo base, sin embargo en clases como el del plástico no se ha logrado superar el resultado al obtener un 16% menos de precisión al igual en vidrio con un 2% menos, con mi dataset, sin embargo en las otras clases se obtiene una mejora considerable. Comparándolo con resultados obtenidos en la literatura, podemos observar que donde se logró una diferencia bastante significativa fue en el modelo realizado con AlexNet [15], llegando a tener una precisión en papel del 83% más,

un valor que no había logrado el modelo de AlexNet, de tal manera se detalla la comparación de resultados en cuanto a precisión en la **tabla 22**, donde se puede observar que ningún modelo ha podido superar mis resultados obtenidos en cuanto a la clase metal y papel, asimismo el único que pudo superar por un 6% mis resultados en cartón fue el modelo MLH-CNN [15], determinando que mi modelo propuesto logra mejores resultados en determinadas clases con tan solo 50 épocas de entrenamiento.

Tabla 22: Comparación de resultados de precisión

Artículos	Dataset	Método	Cartón	Vidrio	Metal	Papel	Plástico
[15]	TrashNet	MLH-CNN	92%	87%	83%	100%	88%
[15]	TrashNet	ResNet50	85%	69%	64%	80%	64%
[15]	TrashNet	Vgg16	85%	69%	54%	71%	72%
[15]	TrashNet	AlexNet	65%	42%	0%	43%	69%
[17]	NWNU-TRASH	DenseNet169	-	84%	80%	78.64%	80.23%
-	TrashNet	Modelo Base	71%	48%	35%	67%	45%
-	Propio	Modelo base	83%	77%	67%	80%	89%
-	TrashNet	Modelo Propuesto	83%	72%	60%	67%	68%
-	Propio	Modelo Propuesto	86%	75%	83%	100%	73%

Elaboración Propia

En cuanto a recall el modelo propuesto logró una mejora en comparación del modelo base, sin embargo en clases como el del vidrio no se ha logrado superar el resultado al obtener un 10% menos de recall, con mi dataset, sin embargo en las otras clases se obtiene una mejora bastante significativa. Comparándolo con resultados obtenidos en la literatura, podemos observar que donde se logró una

diferencia bastante significativa fue en el modelo realizado con AlexNet [15], llegando a tener esa clasificación que no tenía su modelo, encima alcanzando el valor máximo que es el 100%, de tal manera se detalla la comparación de resultados en cuanto a recall en la **Tabla 23**, donde se puede observar que ningún modelo ha podido superar mis resultados obtenidos en cuanto a la clase metal y vidrio, asimismo el único que pudo superar por un 13% mis resultados en plástico fue el modelo MLH-CNN [15], determinando que mi modelo propuesto logra mejores resultados en determinadas clases con tan solo 50 épocas de entrenamiento.

Tabla 23: Comparación de resultados de recall

Artículos	Dataset	Método	Cartón	Vidrio	Metal	Papel	Plástico
[15]	TrashNet	MLH-CNN	94%	79%	97%	93%	93%
[15]	TrashNet	ResNet50	75%	74%	66%	90%	60%
[15]	TrashNet	Vgg16	78%	54%	61%	89%	61%
[15]	TrashNet	AlexNet	80%	54%	0%	78%	43%
[17]	NWNU-TRASH	DenseNet169	-	84%	84%	81%	68.32%
-	TrashNet	Modelo Base	60%	36%	40%	57%	68%
-	Propio	Modelo base	50%	100%	80%	80%	80%
-	TrashNet	Modelo Propuesto	62%	77%	60%	82%	57%
-	Propio	Modelo Propuesto	60%	90%	100%	80%	80%

Elaboración Propia

En cuanto a F1-Score el modelo propuesto logró una mejora en comparación del modelo base, sin embargo en clases como el del vidrio no se ha logrado superar el resultado al obtener un 5% y un 8% en plástico menos de F1-Score, con mi dataset,

sin embargo en las otras clases se obtiene una mejora bastante significativa. Comparándolo con resultados obtenidos en la literatura, podemos observar que donde se logró una diferencia bastante significativa fue en el modelo realizado con AlexNet [15], llegando a tener esa clasificación que no tenía su modelo, encima alcanzando el valor máximo que es el 91%, de tal manera se detalla la comparación de resultados en cuanto a F1-Score en la **Tabla 24**, donde se puede observar que ningún modelo ha podido superar mis resultados obtenidos en cuanto a la clase metal, asimismo el único que pudo superar por un 14% mis resultados en plástico fue el modelo MLH-CNN [15], determinando que mi modelo propuesto logra mejores resultados en determinadas clases con tan solo 50 épocas de entrenamiento.

Tabla 24: Comparación de resultados de F1-Score

Artículos	Dataset	Método	Cartón	Vidrio	Metal	Papel	Plástico
[15]	TrashNet	MLH-CNN	93%	83%	89%	96%	90%
[15]	TrashNet	ResNet50	79%	71%	65%	85%	62%
[15]	TrashNet	Vgg16	81%	61%	57%	79%	66%
[15]	TrashNet	AlexNet	72%	47%	0%	55%	53%
[17]	NWNU-TRASH	DenseNet169	-	84%	81.95%	79.80%	73.80%
-	TrashNet	Modelo Base	65%	41%	37%	62%	54%
-	Propio	Modelo base	62%	87%	73%	80%	84%
-	TrashNet	Modelo Propuesto	71%	74%	60%	74%	62%
-	Propio	Modelo Propuesto	71%	82%	91%	89%	76%

Elaboración Propia

VI. CONCLUSIONES

1. Las pruebas aplicadas, incluida la prueba U de Mann Whitney, respaldan la hipótesis de que el modelo propuesto supera al modelo base de CNN, lo que confirma la validez del enfoque abordado en esta investigación, además de demostrar que mi modelo propuesto y optuna es una estrategia efectiva para mejorar el rendimiento de clasificación de CNN.
2. Asimismo se demuestra que el modelo propuesto logra mejoras en cuanto a precisión, recall y F1-Score en cada una de las clases, además de lograr mejoras en comparación con los modelos elaborados en el estado del arte, logrando un modelo con un buen rendimiento, con capacidad de converger en pocas épocas y con bajo costo computacional, además de no requerir la adición de otros métodos de clasificación.
3. La aplicación de optuna para la optimización de hiperparámetros ha demostrado ser valiosa, al automatizar el proceso de selección de los hiperparámetros. Sin embargo, se observa que esta herramienta puede requerir recursos significativos computacionales y que el resultado final puede depender de la estructura del modelo construido.
4. A pesar de los resultados positivos en esta investigación, se identifican desafíos pendientes, en primer lugar, la necesidad de gestionar de manera eficiente los recursos computacionales debido a que actualmente se ven diferentes modelos de clasificación, sin embargo en un contexto donde la computación intensiva puede ser costosa y poco sostenible esto puede generar un problema, es por ello que se debe buscar un equilibrio entre rendimiento y eficiencia de recursos, además de ello, existe la necesidad de identificar objetos que pueden ser altamente tóxicos en los residuos sólidos clasificados, lo que generaría un problema al momento de reutilizar el producto reciclado.

5. Los resultados de este estudio tienen implicaciones prácticas significativas, particularmente en el contexto de la gestión de residuos y la promoción del reciclaje. Las mejoras en la precisión de la clasificación pueden traducirse en una gestión más eficiente de los recursos para su posterior reciclaje y una reducción de residuos.

VI. RECOMENDACIONES

- Se recomienda la continua investigación de este tipo de modelos con la finalidad de lograr resultados cercanos al 100% de exactitud del modelo, asimismo como lograr los mejores resultados en cuanto a precisión, recall y F1-Score en las clases.
- Asimismo, seguir con la implementación de este algoritmo clasificador en un entorno real, para que pueda ser puesta en funcionamiento y cumpla con el mismo objetivo que tienen este tipo de redes neuronales convolucionales, que es el de automatizar esta clasificación de residuos sólidos para que contribuya de una manera eficiente a la gestión de residuos sólidos.
- Se recomienda a los profesionales y desarrolladores que utilicen Optuna para la optimización de hiper parámetros y que a su vez consideren cuidadosamente la eficiencia de los recursos, ello implicaría realizar un monitoreo y ajuste de los recursos asignados para garantizar un equilibrio adecuado entre la calidad del modelo y la carga computacional.
- A la comunidad académica e investigadores, se recomienda profundizar en el uso de optuna con la finalidad de automatizar el proceso de búsqueda de hiperparametros, para que de esta manera se pueda llegar a la obtención de los mejores hiperparametros para el modelo que construyan y por ende lograr obtener los mejores resultados.
- Para aquellos que implementen este tipo de modelos en entornos prácticos, se sugiere realizar una evaluación rigurosa en diversos conjuntos de datos

representativos de situaciones del mundo real con la finalidad de validar una generalización del modelo y su utilidad en diferentes contextos.

- También se recomienda ampliar la cantidad de imágenes de residuos sólidos domésticos para realizar el entrenamiento en el modelo, mientras la cantidad de imágenes del dataset es mayor el modelo puede generalizar de mejor manera los residuos y de esa forma mejorar los resultados del modelo en cuanto a sus predicciones.
- Se recomienda aplicar este tipo de modelo en aplicaciones móviles para que sean de acceso al público en general, para que de esta manera la clasificación pueda realizarse directamente desde los hogares.

REFERENCIAS BIBLIOGRÁFICAS:

[1] MINAM (2020). Alrededor de 93 mil toneladas de residuos sólidos fueron valorizadas en el 2020 a nivel nacional. Lima, Perú: Ministerio del Ambiente. [en línea]. [fecha de consulta: 1 de mayo del 2023]. <https://www.gob.pe/institucion/minam/noticias/487809-alrededor-de-93-mil-toneladas-de-residuos-solidos-fueron-valorizados-en-el-2020-a-nivel-nacional>

[2] Instituto Nacional de Estadística e Informática. Perú: Estadísticas Ambientales 2021 [en línea]. Lima: Instituto Nacional de Estadística e Informática.[fecha de consulta: 6 de mayo de 2023]. Disponible en: <https://www.gob.pe/institucion/inei/informes-publicaciones/3280963-peru-anuario-de-estadisticas-ambientales-2021>

[3] IBM. (s.f.). What is deep learning? International Business Machines Corporation. [en línea]. [fecha de consulta: 25 de mayo del 2023]. <https://www.ibm.com/topics/deep-learning>

[4] Zapeta, A., Galindo, A., Juan, H. y Martínez, H. (2022). Métricas de rendimiento para evaluar el aprendizaje automático en la clasificación de imágenes petroleras utilizando redes neuronales convolucionales. Ciencia Latina Revista Científica Multidisciplinar, 6(5), 4624. https://doi.org/10.37811/cl_rcm.v6i5.3420

[5] Sivakumar, M, Renuka, P, Chitra, P & Karthikeyan, S. (2022). IoT incorporated deep learning model combined with SmartBin technology for real-time solid waste management. Computational Intelligence. 38, (2), 323–344. <https://doi.org/10.1111/coin.12495>

[6] Wenguang, X., Pengcheng, X., Liguang, Z., Yan, Z., Jinbao, C., Rong, Z. & Yunfeng, X. (2023). Classification and rating of steel scrap using deep learning. Engineering Applications of Artificial Intelligence, 123. <https://doi.org/10.1016/j.engappai.2023.106241>

- [7] Najia, T., Sadia, A., Barna B., Arifa, A. & Riasat K. (2023). Automatic classification of textile visual pollutants using deep learning networks. Alexandria Engineering Journal, 62, 391-402. <https://doi.org/10.1016/j.aej.2022.07.039>
- [8] Melinte, D., Travediu, A. & Dumitriu, D. (2020). Deep Convolutional Neural Networks Object Detector for Real-Time Waste Identification. APPLIED SCIENCES-BASEL, 10(20). <https://doi.org/10.3390/app10207301>
- [9] Ninghui, Y. (2023). Municipal solid waste classification and real-time detection using deep learning methods. Urban Climate, 49. <https://doi.org/10.1016/j.uclim.2023.101462>
- [10] Mookkaiah, S., Thangavelu, G., Hebbar, R., Haldar, N. & Singh, H. (2022). Design and development of smart Internet of Things–based solid waste management system using computer vision. ENVIRONMENTAL SCIENCE AND POLLUTION RESEARCH, 29. <https://doi.org/10.1007/s11356-022-20428-2>
- [11] Perú. Ministerio del Ambiente. Decreto Supremo N° 014-2017-MINAM, Reglamento de la Ley General de Residuos Sólidos. Lima, 7 de junio de 2017.
- [12] Huanping, Z., Hanhua, C., Yuhuai, Z., Changle, G. & Danyu L. (2023). Hybrid deep learning model for accurate classification of solid waste in the society. Urban Climate, 49. <https://doi.org/10.1016/j.uclim.2023.101485>
- [13] Majchrowska, S., Mikołajczyk, A., Ferlin, M., Klawikowska, Z., Plantykw, M., Kwasigroch, A. & Majek, K. (2022). Deep learning-based waste detection in natural and urban environments. Waste Management, 138, 274-284. <https://doi.org/10.1016/j.wasman.2021.12.001>
- [14] Olugboja, Z. (2019). Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network. Procedia Manufacturing. 35, 607-612. <https://doi.org/10.1016/j.promfg.2019.05.086>

- [15] Shi, C., Tan, C., Wang, T. & Wang, L. (2021). A Waste Classification Method Based on a Multilayer Hybrid Convolution Neural Network. *Applied Sciences*, 11, 8572. <https://doi.org/10.3390/app11188572>
- [16] Mitra, A. (2020). Detection of Waste Materials Using Deep Learning and Image Processing. PhD. Tesis, California State University San Marcos, San Marcos, CA, USA, 2020. <https://scholarworks.calstate.edu/downloads/gx41mn74q>
- [17] 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/10.1016/j.wasman.2021.08.038>
- [18] Nnamoko, N., Barrowclough, J. & Procter, J. (2022). Solid Waste Image Classification Using Deep Convolutional Neural Network. 7(4). 2412-3811 <https://doi.org/10.3390/infrastructures7040047>
- [19] Abdella, A., Mofreh, A. & Abd-Allah, M. (2022). *Forecasting of municipal solid waste multi-classification by using time-series deep learning depending on the living standard*. 16. ISSN 2590-1230. <https://doi.org/10.1016/j.rineng.2022.100655>
- [20] Malik, M., Sharma, S., Uddin, M., Chen, C., Wu, C., 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>
- [21] Ziouzos, D., Baras, N., Balafas, V., Dasygenis, M. & Stimoniaris, A. (2022). Intelligent and Real-Time Detection and Classification Algorithm for Recycled Materials Using Convolutional Neural Networks. *Recycling*, 7, 9. <https://doi.org/10.3390/recycling7010009>
- [22] Wahidur, R., Arafat, N. & Mahmud, M. (2022). Intelligent waste management system using deep learning with IoT. *Journal of King Saud University - Computer*

and Information Sciences, 34 (5), 2072-2087.
<https://doi.org/10.1016/j.jksuci.2020.08.016>

[23] Sekar, S. (s.f.). Waste Classification Data, Version 1. [en línea] [Fecha de consulta: 18 de enero de 2022]. <https://www.kaggle.com/techsash/waste-classification-data>

[24] Shahab, S. & Anjum, M. (2022). Solid Waste Management Scenario in India and Illegal Dump Detection Using Deep Learning: An AI Approach towards the Sustainable Waste Management. Sustainability, 14(23).
<https://doi.org/10.3390/su142315896>

[25] Guo D., Yang Q., Zhang Y.-D., Jiang T. & Yan H. (2021). Classification of domestic refuse in medical institutions based on transfer learning and convolutional neural network. CMES - Computer Modeling in Engineering and Sciences, 127 (2), 599 - 620. <https://doi.org/10.32604/cmes.2021.014119>

[26] Patrizi A., Gambosi G. & Zanzotto F. (2021). Data augmentation using background replacement for automated sorting of littered waste. Journal of Imaging, 7 (8). <https://doi.org/10.3390/jimaging7080144>

[27] Arias, F. (2017). Efectividad y eficiencia de la investigación tecnológica en la universidad. RECITIUTM, 3(1). https://www.researchgate.net/profile/Fidias-Arias-Odon-2/publication/320130761_Efectividad_y_eficiencia_de_la_investigacion_tecnologica_en_la_universidad/links/59cf973a4585150177ee1be5/Efectividad-y-eficiencia-de-la-investigacion-tecnologica-en-la-universidad.pdf

[28] Arias, J. & Covinos, M. (2021). Diseño y metodología de la investigación. ENFOQUES CONSULTING EIRL.
<https://repositorio.concytec.gob.pe/handle/20.500.12390/2260>

[29] Vivanco, E. (2020). Gestión de residuos domiciliarios. Asesoría Técnica Parlamentaria.

https://obtienearchivo.bcn.cl/obtienearchivo?id=repositorio/10221/33229/1/BCN_Gestion_de_residuos_domiciliarios_Chile_2022_FINAL.pdf

[30] Yang, J., Chen, B., Gao, B., Liu, J., Li, Y. y Gong, J. (2023). Solid waste mapping based on very high resolution remote sensing imagery and a novel deep learning approach, 38. <https://doi.org/10.1080/10106049.2022.2164361>

[31] Dietrichson, A. (2019). Métodos Cuantitativos. Bookdown. <https://bookdown.org/dietrichson/metodos-cuantitativos/>

[32] Hernandez, J. (2018). Sobre el uso adecuado del coeficiente de correlación de Pearson: definición, propiedades y suposiciones. Archivos Venezolanos de Farmacología y Terapéutica, 37(5). https://www.revistaavft.com/images/revistas/2018/avft_5_2018/25sobre_uso_adecuado_coeficiente.pdf

[33] Manterola, C. et al. (2018). Reliability, precision or reproducibility of the measurements. Methods of assessment, utility and applications in clinical practice. Rev. chil. Infectol, 35(6). 680-688. <http://dx.doi.org/10.4067/S0716-10182018000600680>

[34] Xu, X. et al. (2022). A Data Augmentation Strategy Combining a Modified pix2pix Model and the Copy-Paste Operator for Solid Waste Detection With Remote Sensing Images. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 15, 8484-8491. <https://doi.org/10.1109/JSTARS.2022.3209967>

[35] Verma, V. et. al. (2022). A Deep Learning-Based Intelligent Garbage Detection System Using an Unmanned Aerial Vehicle. Symmetry, 14(5). <https://doi.org/10.3390/sym14050960>

- [36] Althubiti, S. et. al. (2023). Automated biomass recycling management system using modified grey wolf optimization with deep learning model. *Sustainable Energy Technologies and Assessments*, 55, 102936. <https://doi.org/10.1016/j.seta.2022.102936>
- [37] Gómez, M., Danglot, C. y Vega, L. (2003). Sinopsis de pruebas estadísticas no paramétricas. Cuándo usarlas. *Revista Mexicana de Pediatría*, 55(2), pp. 91 - 99. <https://www.ugr.es/~fmocan/MATERIALES%20DOCTORADO/Sinopsis%20de%20pruebas%20estadisticas%20no%20parametricas.pdf>
- [38] Jansi Rani S., Raghu Raman V., Rahul, M. & Prithvi, A. (2022). Multi object detection and classification in solid waste management using region proposal network and YOLO model. *Global Nest Journal*, 24 (4), 743 - 751. <https://doi.org/10.30955/gnj.004501>
- [39] Awe, O., Mengistu, R. & Sreedhar, V. (2017). Smart trash net: Waste localization and classification. arXiv. <http://cs229.stanford.edu/proj2017/final-reports/5226723.pdf>
- [40] Miao, N., Song, Y., Zhou, H. & Li, L. (2020). Do you have the right scissors? Tailoring pre-trained language models via monte-carlo methods. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3436–3441. <https://arxiv.org/abs/2007.06162>

ANEXOS

Anexo 1. Matriz de operacionalización

Variable	Definición Conceptual	Definición Operacional	Dimensiones	Indicadores	Escala de Medición
Variable Dependiente: Clasificación de residuos sólidos domésticos	Los desechos sólidos, también conocidos como basura o residuos, se producen en hogares y diferentes tipos de establecimientos como edificios residenciales, tiendas, restaurantes, hoteles, escuelas y prisiones [29].	Los residuos sólidos domésticos reciclables se clasifican en Papel, Cartón Plástico, Metal y Vidrio [12].	Papel	Precisión Sensibilidad F1-Score	Cuantitativo - Razón
			Cartón		
			Plástico		
			Metal		
			Vidrio		

Anexo 2. Matriz de consistencia

Problema General	Objetivo General	Hipótesis General	Variable	Dimensiones	Indicadores	Metodología
¿En qué medida el modelo de Deep learning de la CNN modificada mejora la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar?	Determinar en qué medida el modelo de Deep learning de la CNN modificada mejora la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar	El modelo de Deep learning de la CNN modificada mejora la clasificación de residuos sólidos domésticos en comparación con el modelo CNN estándar	Variable Independiente: Deep Learning		Exactitud	Tipo de investigación: Aplicada Diseño de la Investigación: Experimental Población: 3000 imágenes de Residuos
Problemas Específicos	Objetivos Específicos	Hipótesis Específicos				

<p>PE1: ¿Qué valor de precisión se obtiene en cada una de las clases de residuos sólidos al utilizar el modelo de Deep learning de la CNN modificada en comparación con el modelo CNN estándar?</p> <p>PE2: ¿Qué valor de sensibilidad se obtiene en cada una de las clases de residuos sólidos al utilizar el modelo de</p>	<p>OE1: Determinar el valor de precisión de cada una de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada en comparación con el modelo CNN estándar</p> <p>OE2: Determinar el valor de sensibilidad de cada una de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep</p>	<p>HE1: El valor de precisión de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar.</p> <p>HE2: El valor de sensibilidad de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN</p>	<p>Variable Dependiente:</p> <p>Clasificación de residuos sólidos domésticos</p>	<p>Papel</p> <p>Cartón</p> <p>Plástico</p> <p>Metal</p>	<p>Precisión</p> <p>Sensibilidad</p> <p>F1-Score</p>	<p>Sólidos Domésticos recolectados de las viviendas del AA. HH. Confraternidad Sector C - Mi Perú</p> <p>Muestra: 100% de la población</p> <p>Técnica de evaluación: Técnica de observación directa</p> <p>Instrumento de evaluación:</p>
--	--	---	---	---	--	--

<p>Deep learning de la CNN modificada en comparación con el modelo CNN estándar?</p> <p>PE3: ¿Qué valor de F1 - Score se obtiene en cada una de las clases de residuos sólidos al utilizar el modelo de Deep learning de la CNN modificada en comparación con el modelo CNN estándar?</p>	<p>learning de la CNN modificada en comparación con el modelo CNN estándar</p> <p>OE3: Determinar el valor de F1 - Score de cada una de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada en comparación con el modelo CNN estándar</p>	<p>modificada logra una mejora en comparación con el modelo CNN estándar</p> <p>HE3: El valor de F1 - Score de alguna de las clases de los residuos sólidos domésticos al utilizar el modelo de Deep learning de la CNN modificada logra una mejora en comparación con el modelo CNN estándar</p>		<p>Vidrio</p>		<p>Ficha de registro de observaciones</p>
--	--	--	--	---------------	--	---

Anexo 3. Fichas de registro

Nombre de la investigación: Deep Learning para la clasificación múltiple de residuos sólidos domésticos Autor: Vilca Quispe, William Frank							PRE-TEST		
							Modelo basado en la Arquitectura CNN estándar		
Ficha de registro									
Modelo de Clasificación de Residuos sólidos domésticos									
N°	Dataset	Clase	Tamaño del Lote	Optimizador	Epochs	Dimensiones de las imágenes	Indicadores		
							Precisión	Recall	F1-Score
1	TrashNet	Cartón	32	adam	30	256 x 256	0,71	0,6	0,65
2		Vidrio				256 x 256	0,48	0,36	0,41
3		Metal				256 x 256	0,35	0,4	0,37
4		Papel				256 x 256	0,67	0,57	0,62
5		Plástico				256 x 256	0,45	0,68	0,54

Nombre de la investigación: Deep Learning para la clasificación múltiple de residuos sólidos domésticos Autor: Vilca Quispe, William Frank	PRE-TEST
	Modelo basado en la Arquitectura CNN estándar

Ficha de registro

Modelo de Clasificación de Residuos sólidos domésticos

N°	Dataset	Clase	Tamaño del Lote	Optimizador	Epochs	Dimensiones de las imágenes	Indicadores		
							Precisión	Recall	F1-Score
1	WasteDom esticNet (Propio)	Cartón	32	adam	30	256 x 256	0,83	0,5	0,62
2		Vidrio				256 x 256	0,77	1	0,87
3		Metal				256 x 256	0,67	0,8	0,73
4		Papel				256 x 256	0,8	0,8	0,8
5		Plástico				256 x 256	0,89	0,8	0,84

Nombre de la investigación: Deep Learning para la clasificación múltiple de residuos sólidos domésticos Autor: Vilca Quispe, William Frank	POST-TEST
	Modelo basado en la Arquitectura CNN (Propio)

Ficha de registro

Modelo de Clasificación de Residuos sólidos domésticos

N°	Dataset	Clase	Tamaño del Lote	Optimizador	Epochs	Dimensiones de las imágenes	Indicadores		
							Precisión	Sensibilidad	F1-Score
1	TrashNet	Cartón	32	Adam	50	256 x 256	0,83	0,62	0,71
2		Vidrio				256 x 256	0,72	0,77	0,74
3		Metal				256 x 256	0,6	0,6	0,6
4		Papel				256 x 256	0,67	0,82	0,74
5		Plástico				256 x 256	0,68	0,57	0,62

Nombre de la investigación: Deep Learning para la clasificación múltiple de residuos sólidos domésticos Autor: Vilca Quispe, William Frank	POST-TEST
	Modelo basado en la Arquitectura CNN (Propio)

Ficha de registro

Modelo de Clasificación de Residuos sólidos domésticos

N°	Dataset	Clase	Tamaño del Lote	Optimizador	Epochs	Dimensiones de las imágenes	Indicadores		
							Precisión	Sensibilidad	F1-Score
1	WasteDomestic Net (Propio)	Cartón	32	Adam	50	256 x 256	0.86	0.60	0.71
2		Vidrio				256 x 256	0.75	0.90	0.82
3		Metal				256 x 256	0.83	1.00	0.91
4		Papel				256 x 256	1.00	0.80	0.89
5		Plástico				256 x 256	0.73	0.80	0.76

Anexo 4. Acuerdo de publicación en revistas de investigación

Yo William Frank Vilca Quispe identificado con DNI N.º 76223277, estudiante de la escuela profesional de Ingeniería de Sistemas de la Universidad César Vallejo - Lima Norte autorizo la divulgación y comunicación pública de mi trabajo de investigación titulado "DEEP LEARNING PARA LA CLASIFICACIÓN MÚLTIPLE DE RESIDUOS SÓLIDOS DOMÉSTICOS"; para la publicación en revistas de investigación académicas, asimismo como el reconocimiento y crédito al autor original, mi persona, en cualquier publicación o reproducción.



FIRMA

DNI: 76223277

FECHA: 15 de octubre del 2023

Anexo 5: Código de la construcción del modelo clasificador

Extracción de las imágenes del dataset Propio

```
import os
import cv2
import numpy as np

main_folder = '/content/soldomestic-net' # Ruta a la carpeta principal
subfolders = ['train', 'validation', 'test']
categories = ['cardboard', 'glass', 'metal', 'paper', 'plastic']

X_train, y_train = [], []
X_val, y_val = [], []
X_test, y_test = [], []

# Corrige el bucle de lectura de imágenes
for subfolder in subfolders:
    for category in categories:
        category_folder = os.path.join(main_folder, subfolder, category)

        if os.path.isdir(category_folder):
            for filename in os.listdir(category_folder):
                file_path = os.path.join(category_folder, filename)

                if os.path.isfile(file_path) and filename.lower().endswith(('.jpg', '.png')):
                    image = cv2.imread(file_path)
                    image = cv2.resize(image, (256, 256))

                    if subfolder == 'train':
                        X_train.append(image)
                        y_train.append(categories.index(category))
                    elif subfolder == 'validation':
                        X_val.append(image)
                        y_val.append(categories.index(category))
                    elif subfolder == 'test':
                        X_test.append(image)
                        y_test.append(categories.index(category))

# Convierte las listas de imágenes y etiquetas en arrays NumPy
X_train = np.array(X_train)
y_train = np.array(y_train)
X_val = np.array(X_val)
y_val = np.array(y_val)
X_test = np.array(X_test)
y_test = np.array(y_test)

# Verificar las formas de los conjuntos de datos
print("Cantidad de imágenes en el conjunto de entrenamiento:", X_train.shape[0])
print("Cantidad de imágenes en el conjunto de validación:", X_val.shape[0])
print("Cantidad de imágenes en el conjunto de prueba:", X_test.shape[0])
```

Extracción de las imágenes del dataset TrashNet

```
import os
import cv2
import numpy as np
from sklearn.model_selection import train_test_split

data_folder = '/content/dataset-resized/'
categories = ['cardboard', 'glass', 'metal', 'paper', 'plastic']

images = []
labels = []

for category in categories:
    category_folder = os.path.join(data_folder, category)

    if os.path.isdir(category_folder):
        for filename in os.listdir(category_folder):
            file_path = os.path.join(category_folder, filename)

            if os.path.isfile(file_path) and any(filename.endswith(extension) for extension in ['.jpg', '.png']):
                image = cv2.imread(file_path)
                image = cv2.resize(image, (256, 256))

                images.append(image)
                labels.append(categories.index(category))

images = np.array(images)
labels = np.array(labels)

# Dividir los datos en conjuntos de entrenamiento, validación y prueba

# random_state = ****
# 100% = 80% / 20%
# 20% = 10% / 10%

X_train, X_val_test, y_train, y_val_test = train_test_split(images, labels, test_size=0.2,
                                                            random_state=42)
X_val, X_test, y_val, y_test = train_test_split(X_val_test, y_val_test, test_size=0.5,
                                                random_state=42)

# Verificar las formas de los conjuntos de datos
total_imagenes = images.shape[0]
print("Total de imágenes en el dataset:", total_imagenes)
print("Cantidad de imágenes en el conjunto de entrenamiento:", X_train.shape[0])
print("Cantidad de imágenes en el conjunto de validación:", X_val.shape[0])
print("Cantidad de imágenes en el conjunto de prueba:", X_test.shape[0])
```

Modelo preparado para los ensayos con optuna

```
def create_model(trial):
    model = models.Sequential()
    model.add(layers.Conv2D(trial.suggest_int('conv1_units', 16, 64), (3, 3), input_shape=(256,
256, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(trial.suggest_int('conv2_units', 16, 64), (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(trial.suggest_int('conv3_units', 16, 64), (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(trial.suggest_int('conv4_units', 16, 64), (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Conv2D(trial.suggest_int('conv5_units', 16, 64), (3, 3), activation='relu'))
    model.add(layers.MaxPooling2D((2, 2)))
    model.add(layers.Dropout(trial.suggest_float('dropout', 0.1, 0.5)))
    model.add(layers.Flatten())
    model.add(layers.Dense(trial.suggest_int('dense_units', 128, 1024), activation='relu', ))
    model.add(layers.Dense(len(categories), activation='softmax'))

    optimizer = Adam(learning_rate=trial.suggest_loguniform('learning_rate', 1e-5, 1e-2))
    model.compile(optimizer=optimizer, loss='categorical_crossentropy', metrics=['accuracy'])

    return model
```

Modelo propuesto

```
import tensorflow as tf
import numpy as np
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import models, layers
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.regularizers import l2

# Convertir las etiquetas en representación one-hot
label_binarizer = LabelBinarizer()
y_train = label_binarizer.fit_transform(y_train)
y_val = label_binarizer.transform(y_val)
y_test = label_binarizer.transform(y_test)

# Normalizar los valores de píxeles entre 0 y 1
X_train = X_train / 255.0
X_val = X_val / 255.0
X_test = X_test / 255.0

# Aplicar aumento de datos a las imágenes de entrenamiento
datagen = ImageDataGenerator(rotation_range=20, width_shift_range=0.1, height_shift_range=0.1,
                             shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
datagen.fit(X_train)

model = models.Sequential()
model.add(layers.Conv2D(38, (3, 3), input_shape=(256, 256, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(54, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(19, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(36, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(30, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Dropout(0.43))
model.add(layers.Flatten())
model.add(layers.Dense(1019, activation='relu', kernel_regularizer=l2(0.001)))
model.add(layers.Dense(len(categories), activation='softmax'))

# Compilar el modelo
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Entrenar el modelo con aumento de datos
history = model.fit(datagen.flow(X_train, y_train, batch_size=32),
                    epochs=50,
                    validation_data=(X_val, y_val))

# Evaluar el modelo en el conjunto de prueba
y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_test, axis=1)
```

Modelo CNN

```
import tensorflow as tf
import numpy as np
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelBinarizer
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras import models, layers
from tensorflow.keras.optimizers import Adam

# Convertir las etiquetas en representación one-hot
label_binarizer = LabelBinarizer()
y_train = label_binarizer.fit_transform(y_train)
y_val = label_binarizer.transform(y_val)
y_test = label_binarizer.transform(y_test)

# Normalizar los valores de píxeles entre 0 y 1
X_train = X_train / 255.0
X_val = X_val / 255.0
X_test = X_test / 255.0

# Aplicar aumento de datos a las imágenes de entrenamiento
datagen = ImageDataGenerator(rotation_range=20, width_shift_range=0.1, height_shift_range=0.1,
                             shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
datagen.fit(X_train)

# Definir la arquitectura del modelo CNN
model = models.Sequential()
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(256, 256, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(len(categories), activation='softmax'))

# Compilar el modelo
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Entrenar el modelo con aumento de datos
history = model.fit(datagen.flow(X_train, y_train, batch_size=32),
                   epochs=30,
                   validation_data=(X_val, y_val))

# Evaluar el modelo en el conjunto de prueba
y_pred = model.predict(X_test)
y_pred = np.argmax(y_pred, axis=1)
y_true = np.argmax(y_test, axis=1)

report = classification_report(y_true, y_pred, target_names=categories)
print(report)
```

Resultados durante el entrenamiento con el modelo propuesto

```
Epoch 1/50
13/13 [=====] - 63s 4s/step - loss: 2.4826 - accuracy: 0.1836 - val_loss: 2.2686 - val_accuracy: 0.2000
Epoch 2/50
13/13 [=====] - 54s 4s/step - loss: 2.1034 - accuracy: 0.2826 - val_loss: 1.9156 - val_accuracy: 0.2600
Epoch 3/50
13/13 [=====] - 53s 4s/step - loss: 1.8142 - accuracy: 0.3551 - val_loss: 1.8105 - val_accuracy: 0.4200
Epoch 4/50
13/13 [=====] - 57s 4s/step - loss: 1.6023 - accuracy: 0.4275 - val_loss: 1.5283 - val_accuracy: 0.4200
Epoch 5/50
13/13 [=====] - 58s 4s/step - loss: 1.3772 - accuracy: 0.4831 - val_loss: 1.4499 - val_accuracy: 0.4000
Epoch 6/50
13/13 [=====] - 56s 4s/step - loss: 1.3332 - accuracy: 0.5435 - val_loss: 1.4877 - val_accuracy: 0.4200
Epoch 7/50
13/13 [=====] - 56s 4s/step - loss: 1.3568 - accuracy: 0.5193 - val_loss: 1.3848 - val_accuracy: 0.4800
Epoch 8/50
13/13 [=====] - 55s 4s/step - loss: 1.2643 - accuracy: 0.5507 - val_loss: 1.4026 - val_accuracy: 0.5000
Epoch 9/50
13/13 [=====] - 54s 4s/step - loss: 1.2837 - accuracy: 0.5266 - val_loss: 1.3479 - val_accuracy: 0.5200
Epoch 10/50
13/13 [=====] - 54s 4s/step - loss: 1.2202 - accuracy: 0.5749 - val_loss: 1.2237 - val_accuracy: 0.5400
Epoch 11/50
13/13 [=====] - 55s 4s/step - loss: 1.1639 - accuracy: 0.5749 - val_loss: 1.2244 - val_accuracy: 0.6400
Epoch 12/50
13/13 [=====] - 55s 4s/step - loss: 1.0966 - accuracy: 0.6377 - val_loss: 1.1645 - val_accuracy: 0.6600
Epoch 13/50
13/13 [=====] - 56s 4s/step - loss: 1.0519 - accuracy: 0.6425 - val_loss: 1.1298 - val_accuracy: 0.6200
Epoch 14/50
13/13 [=====] - 55s 4s/step - loss: 1.0038 - accuracy: 0.6594 - val_loss: 1.0291 - val_accuracy: 0.6600
Epoch 15/50
13/13 [=====] - 54s 4s/step - loss: 0.9541 - accuracy: 0.7005 - val_loss: 0.8514 - val_accuracy: 0.7600
Epoch 16/50
13/13 [=====] - 57s 4s/step - loss: 0.9447 - accuracy: 0.6763 - val_loss: 0.9777 - val_accuracy: 0.6800
Epoch 17/50
13/13 [=====] - 55s 4s/step - loss: 0.8989 - accuracy: 0.7101 - val_loss: 0.9913 - val_accuracy: 0.7000
Epoch 18/50
13/13 [=====] - 55s 4s/step - loss: 0.9350 - accuracy: 0.6836 - val_loss: 0.9126 - val_accuracy: 0.7800
Epoch 19/50
13/13 [=====] - 55s 4s/step - loss: 0.8485 - accuracy: 0.7367 - val_loss: 1.0371 - val_accuracy: 0.6400
Epoch 20/50
13/13 [=====] - 54s 4s/step - loss: 0.9494 - accuracy: 0.6957 - val_loss: 1.0412 - val_accuracy: 0.6800
Epoch 21/50
13/13 [=====] - 57s 4s/step - loss: 0.9297 - accuracy: 0.7126 - val_loss: 0.9845 - val_accuracy: 0.6200
Epoch 22/50
13/13 [=====] - 56s 4s/step - loss: 0.8657 - accuracy: 0.7005 - val_loss: 0.9008 - val_accuracy: 0.6800
Epoch 23/50
13/13 [=====] - 55s 4s/step - loss: 0.8543 - accuracy: 0.6932 - val_loss: 0.8786 - val_accuracy: 0.7000
Epoch 24/50
13/13 [=====] - 54s 4s/step - loss: 0.8061 - accuracy: 0.7101 - val_loss: 0.8204 - val_accuracy: 0.7200
Epoch 25/50
13/13 [=====] - 56s 4s/step - loss: 0.7881 - accuracy: 0.7440 - val_loss: 0.9449 - val_accuracy: 0.6000
Epoch 26/50
13/13 [=====] - 54s 4s/step - loss: 0.8649 - accuracy: 0.6860 - val_loss: 0.8160 - val_accuracy: 0.7400
Epoch 27/50
13/13 [=====] - 58s 4s/step - loss: 0.8546 - accuracy: 0.7343 - val_loss: 0.8992 - val_accuracy: 0.6800
Epoch 28/50
13/13 [=====] - 55s 4s/step - loss: 0.7410 - accuracy: 0.7802 - val_loss: 0.7808 - val_accuracy: 0.7200
Epoch 29/50
13/13 [=====] - 53s 4s/step - loss: 0.7330 - accuracy: 0.7681 - val_loss: 0.8326 - val_accuracy: 0.7200
Epoch 30/50
13/13 [=====] - 53s 4s/step - loss: 0.7225 - accuracy: 0.7585 - val_loss: 0.7259 - val_accuracy: 0.7800
Epoch 31/50
13/13 [=====] - 53s 4s/step - loss: 0.6746 - accuracy: 0.7826 - val_loss: 0.7145 - val_accuracy: 0.8000
Epoch 32/50
13/13 [=====] - 59s 5s/step - loss: 0.7380 - accuracy: 0.7657 - val_loss: 0.7395 - val_accuracy: 0.7800
Epoch 33/50
13/13 [=====] - 53s 4s/step - loss: 0.7513 - accuracy: 0.7826 - val_loss: 0.8523 - val_accuracy: 0.7400
Epoch 34/50
13/13 [=====] - 54s 4s/step - loss: 0.6852 - accuracy: 0.7705 - val_loss: 0.9637 - val_accuracy: 0.7400
Epoch 35/50
13/13 [=====] - 54s 4s/step - loss: 0.6887 - accuracy: 0.7802 - val_loss: 0.7901 - val_accuracy: 0.7800
```

```

Epoch 36/50
13/13 [=====] - 54s 4s/step - loss: 0.6545 - accuracy: 0.8043 - val_loss: 0.7683 - val_accuracy: 0.7800
Epoch 37/50
13/13 [=====] - 56s 4s/step - loss: 0.7273 - accuracy: 0.7681 - val_loss: 0.8730 - val_accuracy: 0.7600
Epoch 38/50
13/13 [=====] - 55s 4s/step - loss: 0.7039 - accuracy: 0.7657 - val_loss: 0.7981 - val_accuracy: 0.7200
Epoch 39/50
13/13 [=====] - 56s 4s/step - loss: 0.6666 - accuracy: 0.7947 - val_loss: 1.0284 - val_accuracy: 0.7000
Epoch 40/50
13/13 [=====] - 54s 4s/step - loss: 0.7542 - accuracy: 0.7609 - val_loss: 0.7271 - val_accuracy: 0.7200
Epoch 41/50
13/13 [=====] - 53s 4s/step - loss: 0.7324 - accuracy: 0.7585 - val_loss: 0.7325 - val_accuracy: 0.8000
Epoch 42/50
13/13 [=====] - 59s 5s/step - loss: 0.6626 - accuracy: 0.8068 - val_loss: 0.8022 - val_accuracy: 0.7800
Epoch 43/50
13/13 [=====] - 54s 4s/step - loss: 0.6629 - accuracy: 0.7947 - val_loss: 0.7104 - val_accuracy: 0.7800
Epoch 44/50
13/13 [=====] - 54s 4s/step - loss: 0.6622 - accuracy: 0.8092 - val_loss: 0.6934 - val_accuracy: 0.8200
Epoch 45/50
13/13 [=====] - 55s 4s/step - loss: 0.6186 - accuracy: 0.8043 - val_loss: 0.8727 - val_accuracy: 0.7600
Epoch 46/50
13/13 [=====] - 55s 4s/step - loss: 0.6199 - accuracy: 0.8213 - val_loss: 0.7377 - val_accuracy: 0.7800
Epoch 47/50
13/13 [=====] - 54s 4s/step - loss: 0.6114 - accuracy: 0.8309 - val_loss: 0.7214 - val_accuracy: 0.7800
Epoch 48/50
13/13 [=====] - 58s 4s/step - loss: 0.5475 - accuracy: 0.8502 - val_loss: 0.6989 - val_accuracy: 0.7800
Epoch 49/50
13/13 [=====] - 57s 4s/step - loss: 0.6769 - accuracy: 0.7826 - val_loss: 0.6884 - val_accuracy: 0.8000
Epoch 50/50
13/13 [=====] - 54s 4s/step - loss: 0.6491 - accuracy: 0.8043 - val_loss: 0.6462 - val_accuracy: 0.7800
2/2 [=====] - 2s 521ms/step

```

Resultados Finales

	precision	recall	f1-score	support
cardboard	0.86	0.60	0.71	10
glass	0.75	0.90	0.82	10
metal	0.83	1.00	0.91	10
paper	1.00	0.80	0.89	10
plastic	0.73	0.80	0.76	10
accuracy			0.82	50
macro avg	0.83	0.82	0.82	50
weighted avg	0.83	0.82	0.82	50

Resultados durante el entrenamiento con el modelo CNN

```
Epoch 1/50
60/60 [=====] - 244s 4s/step - loss: 2.0082 - accuracy: 0.3091 - val_loss: 1.6249 - val_accuracy: 0.4268
Epoch 2/50
60/60 [=====] - 246s 4s/step - loss: 1.5091 - accuracy: 0.4341 - val_loss: 1.3453 - val_accuracy: 0.4854
Epoch 3/50
60/60 [=====] - 240s 4s/step - loss: 1.3074 - accuracy: 0.5282 - val_loss: 1.2521 - val_accuracy: 0.5314
Epoch 4/50
60/60 [=====] - 234s 4s/step - loss: 1.3260 - accuracy: 0.5209 - val_loss: 1.2895 - val_accuracy: 0.5188
Epoch 5/50
60/60 [=====] - 246s 4s/step - loss: 1.2085 - accuracy: 0.5601 - val_loss: 1.2968 - val_accuracy: 0.5146
Epoch 6/50
60/60 [=====] - 239s 4s/step - loss: 1.1650 - accuracy: 0.5863 - val_loss: 1.3142 - val_accuracy: 0.5188
Epoch 7/50
60/60 [=====] - 246s 4s/step - loss: 1.1585 - accuracy: 0.5988 - val_loss: 1.1624 - val_accuracy: 0.5816
Epoch 8/50
60/60 [=====] - 234s 4s/step - loss: 1.0990 - accuracy: 0.6245 - val_loss: 1.2247 - val_accuracy: 0.5774
Epoch 9/50
60/60 [=====] - 232s 4s/step - loss: 1.0557 - accuracy: 0.6412 - val_loss: 1.1484 - val_accuracy: 0.6444
Epoch 10/50
60/60 [=====] - 233s 4s/step - loss: 1.0151 - accuracy: 0.6689 - val_loss: 1.3640 - val_accuracy: 0.5565
Epoch 11/50
60/60 [=====] - 233s 4s/step - loss: 0.9760 - accuracy: 0.6752 - val_loss: 1.3611 - val_accuracy: 0.5690
Epoch 12/50
60/60 [=====] - 233s 4s/step - loss: 0.9702 - accuracy: 0.6799 - val_loss: 1.2713 - val_accuracy: 0.6151
Epoch 13/50
60/60 [=====] - 231s 4s/step - loss: 0.9321 - accuracy: 0.7103 - val_loss: 1.1388 - val_accuracy: 0.6736
Epoch 14/50
60/60 [=====] - 234s 4s/step - loss: 0.8470 - accuracy: 0.7469 - val_loss: 1.2958 - val_accuracy: 0.5941
Epoch 15/50
60/60 [=====] - 232s 4s/step - loss: 0.8555 - accuracy: 0.7505 - val_loss: 1.1586 - val_accuracy: 0.6360
Epoch 16/50
60/60 [=====] - 239s 4s/step - loss: 0.8308 - accuracy: 0.7589 - val_loss: 1.2583 - val_accuracy: 0.6360
Epoch 17/50
60/60 [=====] - 237s 4s/step - loss: 0.8052 - accuracy: 0.7887 - val_loss: 1.2467 - val_accuracy: 0.6569
Epoch 18/50
60/60 [=====] - 231s 4s/step - loss: 0.7795 - accuracy: 0.7950 - val_loss: 1.1712 - val_accuracy: 0.6695
Epoch 19/50
60/60 [=====] - 226s 4s/step - loss: 0.7458 - accuracy: 0.8196 - val_loss: 1.3070 - val_accuracy: 0.6276
Epoch 20/50
60/60 [=====] - 227s 4s/step - loss: 0.7099 - accuracy: 0.8384 - val_loss: 1.3092 - val_accuracy: 0.6820
Epoch 21/50
60/60 [=====] - 226s 4s/step - loss: 0.6954 - accuracy: 0.8353 - val_loss: 1.3056 - val_accuracy: 0.6318
Epoch 22/50
60/60 [=====] - 225s 4s/step - loss: 0.6769 - accuracy: 0.8452 - val_loss: 1.4440 - val_accuracy: 0.6653
Epoch 23/50
60/60 [=====] - 225s 4s/step - loss: 0.6063 - accuracy: 0.8781 - val_loss: 1.4434 - val_accuracy: 0.6778
Epoch 24/50
60/60 [=====] - 225s 4s/step - loss: 0.6132 - accuracy: 0.8771 - val_loss: 1.5419 - val_accuracy: 0.6109
Epoch 25/50
60/60 [=====] - 223s 4s/step - loss: 0.6178 - accuracy: 0.8745 - val_loss: 1.3960 - val_accuracy: 0.6234
Epoch 26/50
60/60 [=====] - 228s 4s/step - loss: 0.5704 - accuracy: 0.8938 - val_loss: 1.4279 - val_accuracy: 0.6402
Epoch 27/50
60/60 [=====] - 224s 4s/step - loss: 0.5808 - accuracy: 0.8902 - val_loss: 1.5051 - val_accuracy: 0.6402
Epoch 28/50
60/60 [=====] - 226s 4s/step - loss: 0.5964 - accuracy: 0.8891 - val_loss: 1.4294 - val_accuracy: 0.6946
Epoch 29/50
60/60 [=====] - 228s 4s/step - loss: 0.5352 - accuracy: 0.9184 - val_loss: 1.4828 - val_accuracy: 0.7029
Epoch 30/50
60/60 [=====] - 231s 4s/step - loss: 0.5762 - accuracy: 0.8996 - val_loss: 1.4454 - val_accuracy: 0.6987
Epoch 31/50
60/60 [=====] - 229s 4s/step - loss: 0.5268 - accuracy: 0.9195 - val_loss: 1.4012 - val_accuracy: 0.6862
Epoch 32/50
60/60 [=====] - 225s 4s/step - loss: 0.5380 - accuracy: 0.9132 - val_loss: 1.6903 - val_accuracy: 0.6025
Epoch 33/50
60/60 [=====] - 227s 4s/step - loss: 0.5668 - accuracy: 0.9064 - val_loss: 1.3974 - val_accuracy: 0.6653
Epoch 34/50
60/60 [=====] - 224s 4s/step - loss: 0.5281 - accuracy: 0.9184 - val_loss: 1.5685 - val_accuracy: 0.6569
Epoch 35/50
60/60 [=====] - 225s 4s/step - loss: 0.4933 - accuracy: 0.9378 - val_loss: 1.6449 - val_accuracy: 0.6695
```



```

Epoch 36/50
60/60 [=====] - 225s 4s/step - loss: 0.4358 - accuracy: 0.9535 - val_loss: 1.6791 - val_accuracy: 0.6611
Epoch 37/50
60/60 [=====] - 226s 4s/step - loss: 0.4867 - accuracy: 0.9315 - val_loss: 1.5502 - val_accuracy: 0.6485
Epoch 38/50
60/60 [=====] - 226s 4s/step - loss: 0.5087 - accuracy: 0.9289 - val_loss: 1.4556 - val_accuracy: 0.6820
Epoch 39/50
60/60 [=====] - 223s 4s/step - loss: 0.4588 - accuracy: 0.9467 - val_loss: 1.5844 - val_accuracy: 0.6778
Epoch 40/50
60/60 [=====] - 225s 4s/step - loss: 0.4967 - accuracy: 0.9357 - val_loss: 1.4094 - val_accuracy: 0.7155
Epoch 41/50
60/60 [=====] - 226s 4s/step - loss: 0.4303 - accuracy: 0.9529 - val_loss: 1.6524 - val_accuracy: 0.6904
Epoch 42/50
60/60 [=====] - 221s 4s/step - loss: 0.4441 - accuracy: 0.9482 - val_loss: 1.6001 - val_accuracy: 0.6653
Epoch 43/50
60/60 [=====] - 226s 4s/step - loss: 0.4642 - accuracy: 0.9367 - val_loss: 1.8574 - val_accuracy: 0.6192
Epoch 44/50
60/60 [=====] - 228s 4s/step - loss: 0.5114 - accuracy: 0.9236 - val_loss: 1.6456 - val_accuracy: 0.6904
Epoch 45/50
60/60 [=====] - 227s 4s/step - loss: 0.4915 - accuracy: 0.9257 - val_loss: 1.5110 - val_accuracy: 0.7280
Epoch 46/50
60/60 [=====] - 226s 4s/step - loss: 0.4532 - accuracy: 0.9467 - val_loss: 1.4841 - val_accuracy: 0.6695
Epoch 47/50
60/60 [=====] - 227s 4s/step - loss: 0.4265 - accuracy: 0.9514 - val_loss: 1.6442 - val_accuracy: 0.6485
Epoch 48/50
60/60 [=====] - 228s 4s/step - loss: 0.4459 - accuracy: 0.9461 - val_loss: 1.5213 - val_accuracy: 0.6862
Epoch 49/50
60/60 [=====] - 222s 4s/step - loss: 0.4901 - accuracy: 0.9263 - val_loss: 1.5467 - val_accuracy: 0.6611
Epoch 50/50
60/60 [=====] - 219s 4s/step - loss: 0.4763 - accuracy: 0.9399 - val_loss: 1.5127 - val_accuracy: 0.7238
8/8 [=====] - 7s 847ms/step

```

Resultados Finales

	precision	recall	f1-score	support
cardboard	0.83	0.62	0.71	40
glass	0.72	0.77	0.74	56
metal	0.60	0.60	0.60	43
paper	0.67	0.82	0.74	56
plastic	0.68	0.57	0.62	44
accuracy			0.69	239
macro avg	0.70	0.68	0.68	239
weighted avg	0.70	0.69	0.69	239



UNIVERSIDAD CÉSAR VALLEJO

**FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

Declaratoria de Autenticidad del Asesor

Yo, NECOCHEA CHAMORRO JORGE ISAAC, docente de la FACULTAD DE INGENIERÍA Y ARQUITECTURA de la escuela profesional de INGENIERÍA DE SISTEMAS de la UNIVERSIDAD CÉSAR VALLEJO SAC - LIMA NORTE, asesor de Tesis titulada: "DEEP LEARNING PARA LA CLASIFICACIÓN MÚLTIPLE DE RESIDUOS SÓLIDOS DOMÉSTICOS", cuyo autor es VILCA QUISPE WILLIAM FRANK, constato que la investigación tiene un índice de similitud de 15.00%, verificable en el reporte de originalidad del programa Turnitin, el cual ha sido realizado sin filtros, ni exclusiones.

He revisado dicho reporte y concluyo que cada una de las coincidencias detectadas no constituyen plagio. A mi leal saber y entender la Tesis cumple con todas las normas para el uso de citas y referencias establecidas por la Universidad César Vallejo.

En tal sentido, asumo la responsabilidad que corresponda ante cualquier falsedad, ocultamiento u omisión tanto de los documentos como de información aportada, por lo cual me someto a lo dispuesto en las normas académicas vigentes de la Universidad César Vallejo.

LIMA, 27 de Noviembre del 2023

Apellidos y Nombres del Asesor:	Firma
NECOCHEA CHAMORRO JORGE ISAAC DNI: 18167347 ORCID: 0000-0002-3290-8975	Firmado electrónicamente por: JNECOCHEA el 28- 11-2023 09:22:02

Código documento Trilce: TRI - 0667048