

# Clasificador de residuos : Predicción tipo de material de residuos

Elías Gangas Donoso<sup>1</sup>

<sup>1</sup>Departamento de Matemática y Ciencia de la Computación, Universidad de Santiago de Chile, Santiago, Chile

## 1 Resumen

En este informe se presenta un proyecto de inteligencia artificial el cual consiste en el entrenamiento de tres modelos de redes neuronales convolucionales (CNN) para la clasificación de imágenes. Se describe el procedimiento de entrenamiento del modelo, y los resultados obtenidos en términos de precisión y rendimiento.

El modelo que obtuvo la mayor precisión se seleccionó para su implementación en una página web interactiva. Esta implementación incluye el desarrollo de una API utilizando Flask para recibir y procesar las imágenes enviadas por los usuarios, y devolver las predicciones en formato JSON.

Además, se detallan las tecnologías y herramientas utilizadas, como TensorFlow y Keras para el entrenamiento de los modelos. Se discuten las mejoras realizadas durante el desarrollo, los desafíos enfrentados, y las posibles direcciones para trabajos futuros.

## 2 Problemática

La clasificación de residuos es un proceso esencial para la gestión eficiente de desechos, la reducción del impacto ambiental y la promoción del reciclaje. La problemática en la clasificación de residuos abarca varios aspectos, entre los cuales se incluyen:

- **Variedad de residuos:** Los residuos pueden ser de diversos tipos, como orgánicos, vidrios, papel, cartón, entre otros. La gran variedad de residuos dificulta su identificación y clasificación correcta.
- **Dificultades en la Clasificación Manual:** La clasificación manual, aunque efectiva, es una tarea tediosa y propensa a errores humanos. Además, los trabajadores expuestos a residuos pueden enfrentar condiciones insalubres y peligrosas.
- **Infraestructura Insuficiente:** La falta de centros de reciclaje bien equipados y accesibles dificulta la correcta gestión de residuos. Implementar sistemas de recolección selectiva puede ser un desafío en áreas rurales o de bajos ingresos.

Por estas razones, es crucial implementar un modelo de clasificación eficaz que logre identificar y clasificar correctamente los diferentes tipos de residuos

## 3 Solución Propuesta

La solución propuesta para este problema es entrenar un modelo de clasificación de imágenes de residuos utilizando redes neuronales convolucionales (CNN). El proceso incluye los siguientes pasos:

- **Recolección de Datos:** Recolectar un conjunto de datos (dataset) que contenga imágenes de distintos tipos de residuos, como papel, cartón, plástico, entre otros.
- **Entrenamiento del Modelo:** Utilizar el dataset para entrenar una red neuronal convolucional, optimizando el modelo para alcanzar un alto porcentaje de precisión en la clasificación de residuos.
- **Despliegue en una Página Web:** Implementar el modelo entrenado en una página web a través de una API creada con Flask. La API permitirá que los usuarios suban imágenes de residuos desde sus dispositivos.

Con esta solución, se obtendrá un clasificador de imágenes de residuos con un alto porcentaje de acierto. Los usuarios podrán subir imágenes de residuos desde sus dispositivos, y el sistema las clasificará automáticamente, facilitando así la gestión eficiente de residuos.

### 3.1 Procedimiento

El primer paso fue encontrar un dataset adecuado para la clasificación de imágenes de residuos. Inicialmente, se utilizó el dataset RealWaste, obtenido de UCI (Irvine Machine Learning Repository). Sin embargo, debido a la escasez de imágenes (2,527 imágenes), el modelo no entrenaba adecuadamente. Para solucionar esto, se combinó con otro dataset llamado waste-classification-data de Kaggle.

De estos datasets se creó uno nuevo, dividido en tres carpetas: train, validation y test. Cada carpeta contiene 1000 imágenes por clase, distribuidas de la siguiente manera: 600 para entrenamiento, 200 para validación y 200 para prueba. Dentro de estas carpetas, las imágenes se organizan en subcarpetas con el nombre de las clases de los residuos correspondientes. Se realizó una revisión al nuevo dataset para asegurar que todas las imágenes fueran claras y adecuadas para el entrenamiento.

Para el entrenamiento del modelo, se creó un notebook (.ipynb) en Google Colab. Sin embargo, el hardware asignado por defecto no fue suficiente, por lo que se decidió

utilizar Jupyter Notebook en un entorno local con Python 3.7, aprovechando el hardware disponible y todas las librerías necesarias.

En el entorno creado se utilizaron las siguientes librerías:

- CUDA 10.1
- cuDNN 10.1
- TensorFlow 2.3
- Keras 2.3
- Numpy
- PIL
- Matplotlib
- sklearn
- Opencv

3.2 Hardware

Table 1: Especificaciones del Hardware

Componente	Especificación
Procesador	Ryzen 5 3600
RAM	32 GB DDR4
Tarjeta Gráfica	NVIDIA GeForce 1660 Ti
Almacenamiento	SSD 500 MB
Sistema Operativo	Windows 10

3.3 Entrenamiento

Para el entrenamiento, se leyó el dataset con sus respectivas clases (cardboard, glass, metal, organic, paper, plastic, trash) y se realizó una aumentación de datos para incrementar la variabilidad de las imágenes, permitiendo así entrenar un modelo más robusto. Se probaron tres modelos y se seleccionó el que mejor rendimiento mostró en términos de precisión.



Figure 1: Aumentacion de datos

Finalmente, se procedió a desarrollar una página web utilizando Flask, HTML, CSS y JavaScript para presentar el modelo y conectarlo correctamente. La página web permite a los usuarios subir imágenes de residuos desde sus dispositivos, las cuales son clasificadas automáticamente por el modelo.

4 Resultados

4.1 Entrenamiento Modelos

Comenzando con el desarrollo del entrenamiento, se entrenaron tres modelos VGG16, InceptionResNetV4 y DenseNet121.

El modelo que menos demoró en entrenar fue VGG16, aproximadamente unos 40 minutos, mientras que los otros modelos, InceptionResNetV4 y DenseNet121, requerían cerca de 90 minutos cada uno, incluyendo su ajuste fino que también tomaba un tiempo similar. Con todos los ajustes necesarios se obtuvieron los siguientes resultados:

- **VGG16:** Este modelo obtuvo una precisión de 0.9992 (99.92%). Sin embargo, al observar su matriz de confusión, se puede ver que sus predicciones no son consistentes, lo que llevó a descartar este modelo.

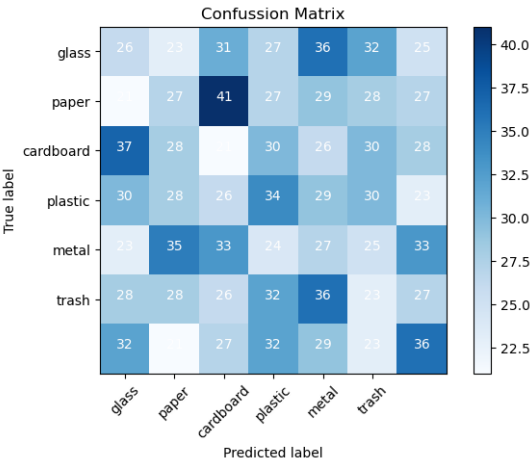


Figure 2: Matriz de confusión para VGG16

- **InceptionResNetV4:** Este modelo obtuvo un accuracy de 0.8790 (88%). Después de realizar un ajuste fino, se logró aumentar su precisión a 0.9164 (92%).

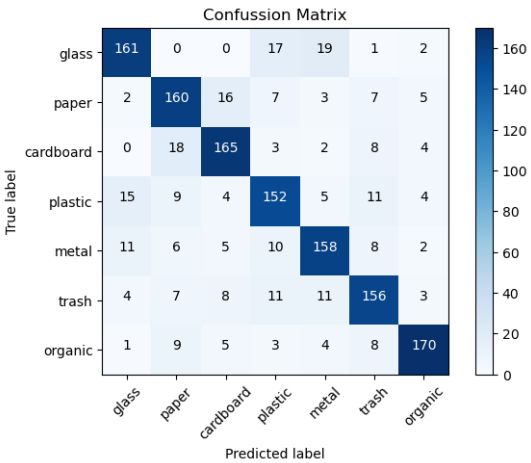
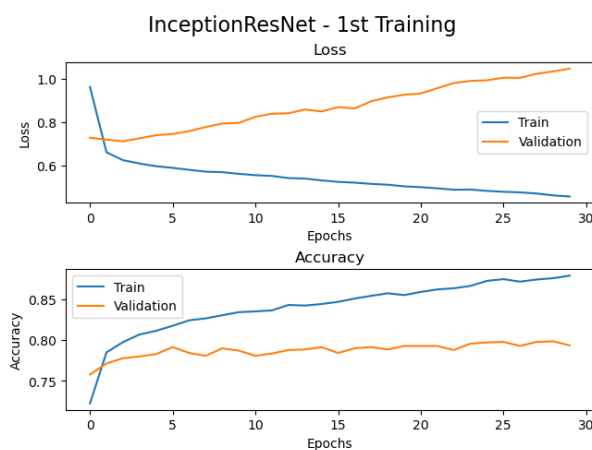
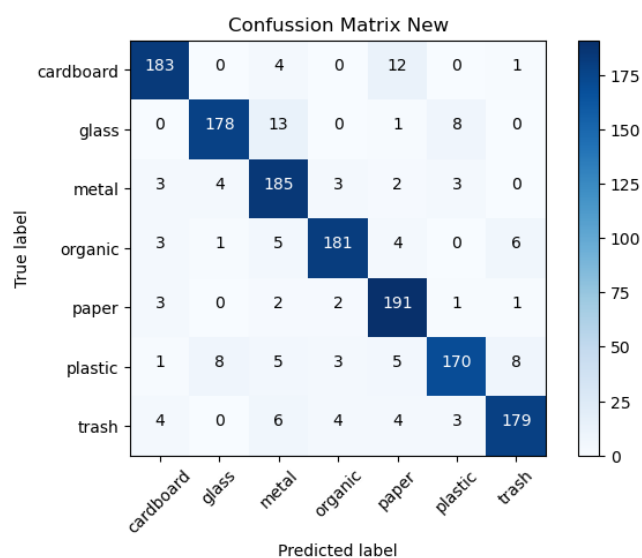


Figure 3: Matriz de confusión para InceptionResV4  
Observando su matriz de confusión podemos ver que predice con algunos errores

– Primer entrenamiento:



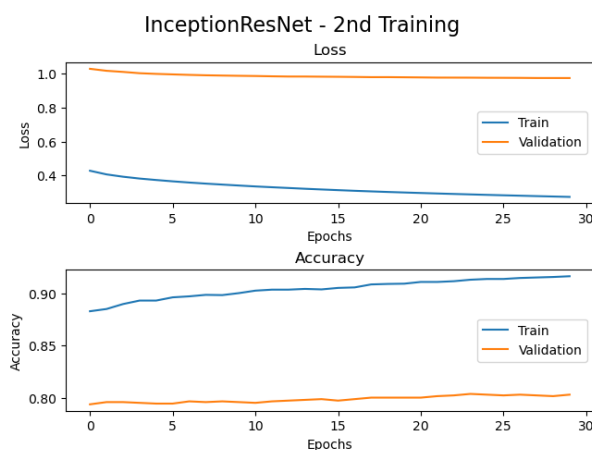
**Figure 4:** Primer entrenamiento Densenet121



**Figure 6:** Matriz de confusión para Densenet121

Como podemos observar, el modelo que mejor predice según su matriz de confusión es el modelo preentrenado DenseNet121. Por lo tanto, se utilizó este modelo para el resto del desarrollo.

– Ajuste fino:



**Figure 5:** Ajuste fino Densenet121

Observando los graficos de entrenamiento podemos ver que luego del ajuste fino su grafica cambia, esto posiblemente se deba a un sobreajuste del modelo

- **DenseNet121:** Este modelo obtuvo inicialmente una precisión de 0.9019 (90%). Tras un ajuste fino, se mejoró su precisión a 0.9295 (93%). Observando su matriz de confusión, se puede ver que es el modelo que mejor predice de los tres, por lo que se seleccionó para el proyecto.

– Primer entrenamiento:

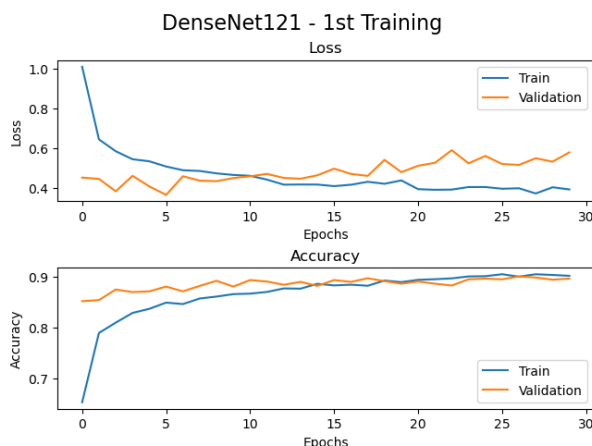


Figure 7: Primer entrenamiento Densenet121



Figure 9: Prediccion botella vidrio

– Ajuste fino:

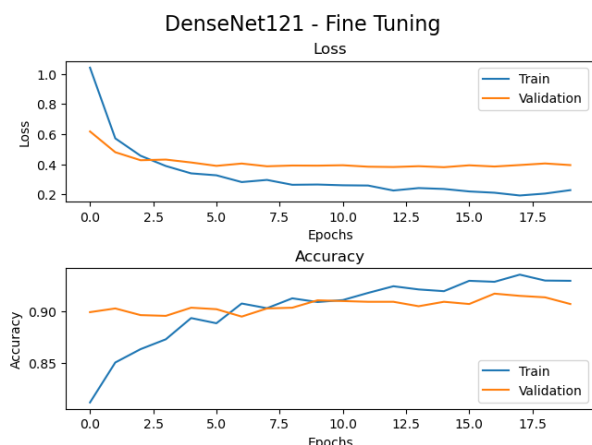


Figure 8: Ajuste fino Densenet121

Luego de comprobar el funcionamiento del modelo, se añadió una capa adicional de preprocesamiento para que las imágenes se reajusten y normalicen antes de ser procesadas por el modelo. Al analizar la nueva matriz de confusión, se comprobó que no existe diferencia significativa con la anterior, por lo que la adición de la capa no afectó la precisión del modelo.

Finalmente, el modelo se testeó con imágenes propias contenidas en la carpeta Custom, confirmando así su capacidad para clasificar correctamente los tipos de residuos en condiciones reales. El modelo predice correctamente pero tuvo problemas con aquellos que pudieran considerarse similares como cartón y papel, o metal y vidrio tal como se muestra en su matriz de confusión.

## 4.2 Implementacion web

Para la implementación web, se levantó un servidor utilizando Flask para conectar el modelo obtenido anteriormente. Además, se utilizó Bootstrap junto con archivos HTML, CSS y JavaScript para lograr un diseño responsive y amigable para el usuario.

Comenzando con Flask, se configuró el servidor para que pueda recibir solicitudes de imágenes desde la interfaz web y utilizar el modelo DenseNet121 preentrenado para clasificar los residuos correctamente. Esta integración permite que los usuarios carguen imágenes desde sus dispositivos, las cuales son procesadas por el modelo y muestran el resultado de la clasificación de residuos en la página.

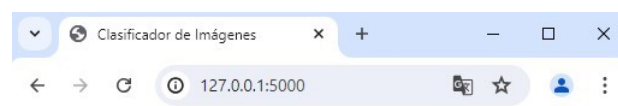


Figure 10: Diseño inicio pagina



Figure 11: Diseño seleccion



Figure 12: Diseño prediccion

5 Conclusiones

En resumen, este proyecto ha abordado de manera efectiva la clasificación de residuos mediante el entrenamiento y selección de un modelo avanzado de aprendizaje profundo, específicamente el DenseNet121, utilizando TensorFlow y Keras. Este modelo demostró una alta precisión en la clasificación de diferentes tipos de residuos, proporcionando una base sólida para aplicaciones prácticas.

La implementación web a través de Flask permitió conectar el modelo con una interfaz accesible, donde los usuarios pueden cargar imágenes de residuos para obtener clasificaciones instantáneas. El diseño responsive logrado con Bootstrap, junto con el uso de HTML, CSS y JavaScript, aseguró una experiencia de usuario intuitiva y atractiva.

Mirando hacia el futuro, una mejora potencial sería la integración de técnicas de segmentación de imágenes. Esto permitiría no solo clasificar los residuos, sino también identificar y delimitar áreas específicas de los objetos, lo que sería fundamental para la automatización en aplicaciones como robots recolectores o brazos mecánicos en plantas de reciclaje. Esto optimizaría aún más el proceso de manejo de residuos, aumentando la eficiencia y reduciendo la dependencia de la intervención humana.

En conclusión, este proyecto no solo demuestra el poder de las redes neuronales convolucionales en la clasificación de imágenes, sino también su potencial para mejorar sistemas automatizados que contribuyen a la sostenibilidad ambiental y la gestión eficiente de residuos..