

**Desarrollo de un Algoritmo Computacional basado en Deep Learning para la Clasificación
de Residuos Orgánicos**

Zalma Valentina Moreno Galeano

Asesores,

Miguel Ángel Vargas Valencia, PhD.

Freddy Alexander Torres Payoma, PhD(c).

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas, Tecnología e Ingeniería ECBTI

Ingeniería de Sistemas

2023

Nombre Director de Trabajo de Grado

Jurado

Jurado

Agradecimientos

El camino hacia esta tesis ha estado lleno de desafíos y momentos de aprendizaje, y no habría sido posible sin el apoyo inquebrantable de las personas que han estado a mi lado. Quiero expresar mi profundo agradecimiento a mi familia, quienes han sido mi fuente constante de inspiración y aliento. A mis padres Nubia Esperanza y Jorge Eduardo, quienes me han brindado amor incondicional, orientación y sacrificio a lo largo de mi vida. Vuestra dedicación y confianza en mí me han impulsado a alcanzar este hito académico.

A mis hermanos Joan Esteban y Jadde Violeta, quienes han compartido mis alegrías y desafíos, y cuyas palabras de aliento me han dado la fuerza necesaria para seguir adelante. Sus risas y apoyo constante han sido un faro de luz en mi camino.

No puedo dejar de mencionar a mis familiares, amigos cercanos, quienes han estado a mi lado en cada paso del camino. Sus palabras de aliento, sus consejos y su compañía han hecho que este viaje sea mucho más significativo. Cada conversación y momento compartido ha sido una fuente de motivación y felicidad.

A mis profesores Miguel Vargas, PhD. y el profesor Freddy Torres Payoma, PhD(c)., quienes han dedicado su tiempo y conocimiento para guiarme. Sus orientaciones y sugerencias han enriquecido mi trabajo y han contribuido al desarrollo de mis habilidades académicas.

Resumen

El proyecto desarrolla y diseña un modelo de inteligencia artificial para visión computacional basado en redes neuronales convolucionales que permite la clasificación de residuos orgánicos, en dos categorías previamente establecidas, a saber: categoría 1. residuos orgánicos reutilizables y categoría 2. residuos orgánicos compostables. El proceso inicia con la recolección de imágenes provenientes de bases de datos existentes e imágenes capturadas en la finca Carrizalito del municipio de Sopó. Posteriormente se realiza el entramiento del sistema mediante el uso de lenguaje de programación en Python usando paquetes informáticos como lo son TensorFlow y Keras. Finalmente, se evalúa el modelo mediante diferentes métricas de aprendizaje de máquina para su validación.

Palabras clave: Inteligencia artificial, aprendizaje de máquina, visión computacional, red neuronal convolucional.

Abstract

The Project develops and designs an artificial intelligence model for computer vision based on convolutional neural networks that allows the classification of organic waste into two previously established categories, namely: Category 1. Reusable organic waste and Category 2. Compostable organic waste. The process begins with the collection of images from existing databases and images captured on Carrizalito farm in the municipality of Sopó. Subsequently, the system is trained through the use Python programming language using computer packages such as TensorFlow and Keras. Finally, the model is evaluated using different machine learning metrics for validation.

Keywords: Artificial intelligence, machine learning, computer vision, convolutional neuronal network.

Tabla de Contenido

Problemas de Investigación.....	11
Introducción.....	11
Antecedentes del Problema de Investigación	12
Formulación del Problema.....	16
Justificación.....	19
Objetivos	21
Objetivo General.....	21
Objetivo Especifico	21
Metodología.....	22
Fundamentación Teórica	25
Análisis de Clasificación de Residuos Orgánicos	25
Inteligencia Artificial	30
Aprendizaje Automático.....	31
Aprendizaje Supervisado	32
Aprendizaje no Supervisado	33
Aprendizaje Profundo	34
Redes Neuronales Artificiales.....	34
RGB	35
Convolución.....	37
Capas de una Red Neuronal.....	39

Capa de Convolución	39
Capa de Pooling	40
Proceso de Submuestreo.....	41
Funciones de Activación	42
Aumento de Imágenes	46
Pesos de Clase.....	47
Class Weight for Inbalanced Dataset	48
Métricas para Clasificadores de Aprendizaje Automático	48
Sobreajuste	50
Técnicas de Dropout.....	50
Arquitecturas CNN.....	51
Redes Neuronales Convoluciones Profundas	54
Herramientas y Librerías	55
Algoritmo Basado en Deep Learning para la Clasificación de Residuos Orgánicos	61
Análisis de Resultados.....	68
Conclusiones.....	75
Referencias Bibliográficas.....	77

Lista de Figuras

Figura 1 Proceso de análisis de datos en el aprendizaje automático.....	30
Figura 2 Vista de una de las correlaciones en este caso las variables Humedad del suelo y Temperatura	32
Figura 3 Grafica de cubo RGB realizada en GeoGebra	34
Figura 4 Imagen desde una perspectiva de espacio de color RGB.....	35
Figura 5 Imagen en escala de grises y la muestra de una fracción de la imagen como pixel.....	36
Figura 6 Nueva imagen por medio de la original dado el concepto de kernel	37
Figura 7 Capa de relleno de ceros añadidos a la imagen original.....	39
Figura 8 Ilustración de Max Pooling.....	40
Figura 9 Grafica de la función de activación Sigmoid en el plano 2D.	44
Figura 10 Aumento de datos de las imágenes de residuos orgánicos clasificados entre reutilizable y de compostaje.....	45
Figura 11 Matriz de confusión. Nota. VP = Positivo verdadero, FN = Falso Negativo, FP = Falso Positivo, VN = Verdadero Negativo.....	48
Figura 12 Se representa el esquema que conlleva una imagen en la red neuronal convolucional, la cual pasa por las capas de entrada, ocultas y salida, donde en la salida ya sale con la clasificación de imagen, ya sea reutilizable o de compostaje	54
Figura 13 Proceso de imagen luego de usar Matplotlib	56
Figura 14 Ejemplo del Dataset – a. Residuos orgánicos reutilizables, b. Residuos orgánicos Compostaje	58

Figura 15 Diseño del sistema mecánico de clasificación de residuos orgánicos.....	60
Figura 16 Primeras pruebas del algoritmo y sus resultados en cuanto a su precisión y perdida del modelo	69
Figura 17 Modelo corrido con arquitectura VGG19 a 10 épocas.....	70
Figura 18 Algunas de las arquitecturas CNN usando el modelo y corriendo a 50 épocas	71
Figura 19 Arquitecturas MobilNetV2 y VGG19 específicamente, corridas con 100 épocas	72

Lista de Tablas

Tabla 1 Un filtro de tamaño 3×3	37
---	----

Problemas de Investigación

Introducción

La integración de técnicas de visión por computadora y aprendizaje automático en la agricultura es fundamental para satisfacer la creciente demanda de alimentos, tal como se destaca en el libro Uddin y Basal (2021), la necesidad de aumentar la eficiencia en la producción de cultivos de manera sostenible y rentable se ha vuelto cada más urgente debido al incremento de la población y la urbanización.

El libro de Kaza *et al.* (2018) nos dice que se espera un aumento del 70% en la cantidad de desechos a nivel global para el año 2050. A menos que tomemos medidas urgentes para establecer un control efectivo en la gestión final de los residuos sólidos, esta previsión podría volverse realidad incluso antes debido a la falta de una supervisión adecuada en los lugares donde se desechan estos materiales. Uno de los factores que contribuyen a este incremento en la acumulación de desechos en los sitios de disposición final es el crecimiento poblacional. Por eso, se han implementado diversas medidas legales con el objetivo de regular los impactos que los desechos generan, manteniendo al mismo tiempo un equilibrio para no poner en peligro el medio ambiente ni la salud pública.

En este contexto, la investigación en visión por computadora y aprendizaje automático ha experimentado un notable crecimiento, con el propósito de mejorar la eficiencia de producción agrícola y reemplazar el trabajo intensivo en mano de obra. Los algoritmos de aprendizaje automático, respaldados por el poder de las unidades de procesamiento gráfico (GPU), permiten el análisis de grandes volúmenes de datos complejos. Esto ha posibilitado el desarrollo de modelos computacionales con una capacidad de procesamiento paralelo significativa,

transformando así la forma en que se abordan los desafíos de la producción de alimentos a gran escala.

Las técnicas de visión por computadora y aprendizaje automático en el entorno de GPU han encontrado una amplia gama de aplicación en tareas agrícolas diversas abarcando el procesamiento de numerosas imágenes adquiridas por humanos, robots, drones y otros sensores remotos. Estos enfoques han sido fundamentales para impulsar la sostenibilidad agrícola, al permitir un monitoreo más preciso y oportuno de los cultivos, así como una mejor comprensión de las condiciones ambientales que afectan su desarrollo.

Antecedentes del Problema de Investigación

En esta sección de antecedentes se ofrece una visión completa de los estudios y trabajos previos relacionados con la clasificación de residuos orgánicos y las redes neuronales artificiales.

Se encuentra una serie de estudios destacados que han abordado la clasificación de residuos utilizando diferentes enfoques y técnicas de aprendizaje automático, estos por los cuales han sido escogidos de Ojeda, ya que en cuestión el documento representa una revisión sistemática de la literatura sobre la clasificación de residuos sólidos utilizando técnicas de aprendizaje automático. Se detallan las diversas técnicas desarrolladas a nivel internacional para la clasificación de residuos orgánicos mediante el uso de técnicas de aprendizaje de aprendizaje automático de máquina. La metodología empleada en el estudio sigue lo lineamientos de PRISMA, proporcionando una sólida base bibliográfica de investigación.

Para hablar sobre redes neuronales artificiales y su integración en el tema de agricultura se toma información documentada de Uddin y Basal (2021), el cual discute los avances recientes, modificaciones y aplicaciones de técnicas de aprendizaje automático, inteligencia artificial y sistemas inteligentes en áreas relacionadas con la agricultura, como la detección de enfermedades en plantas y la clasificación de insectos. También proporciona con información detallada sobre la arquitectura de redes neuronales convolucionales y su aplicación como por ejemplo en la detección de enfermedades en hojas de tomate.

Con esta base sólida se podrá abordar el problema de clasificación de residuos mediante inteligencia artificial y nos inspira a buscar soluciones innovadoras en este campo.

Estudios Referentes a la Clasificación de Residuos Orgánicos

- Nnamoko *et al.* (2022) dieron como objetivo principal, presentar una arquitectura de redes neuronales convoluciones para la clasificación de residuos, sus resultados presentados fueron antes el modelo de línea de base y de esa misma manera sobre la arquitectura CCN, con una resolución de imágenes grande y pequeñas.
- Altikat *et al.* (2022) hicieron un análisis de redes neuronales convolucionales profundas DCNN para obtener los porcentajes basado en la estructura observadas en la red bajo 120 observaciones, lo que mantuvieron sus resultados en que el DCNN5 se tuvo una muestra de la predicción del 76.7% de materia organiza versus un DCNN4 con un 83% de materia orgánica.

- Tao *et al.* (2021) se centran en la toma de imágenes de buena calidad GF-1 B/D de residuos de cultivos en china para luego poder realizar con la ayuda de un estudio con el método de aprendizaje profundo mapear áreas que se concentren en lo residuos de maíz. Dentro del proceso de desarrolla una red MSCU-net lo que se tiene en cuenta el agrupamiento de convoluciones multiescalar, lo que al tenerse se presenta como ayuda para la construcción del MSCU-net+C y lograr una mejor optimización de las imágenes resultantes.
- Guo *et al.* (2021) realiza un análisis sistemático de la aplicación que ha tenido la investigación de los modelos de aprendizaje profundo en el campo de la agricultura especialmente en los residuos sólidos orgánicos en los tiempos establecidos 2003 y 2020. Se tuvo como resultados que la mayoría de los estudios fueron realizados en municipios donde hacen un manejo de residuos orgánicos.
- Qiu *et al.* (2020) se buscaba un modelo de clasificación el cual pudiera mirar los niveles de contaminación plástica en el suelo, ya sean en diferentes sectores con esa contaminación dada. Como resultados se pudo adquirir modelos transferibles que lograban la buena clasificación, dentro de los métodos más nombras y eficientes fue el MEDA.
- Toğaçar *et al.* (2020) dado como objetivo la aplicación del método Ridge Regression (RR) por medio de una implementación de máquinas de vectores de soporte se dan como resultados la exitosa clasificación de residuos dando así un índice de precisión del 99,95%.
- Vo *et al.* (2019) propone un modelo de red neuronal convolucionada llamada RRGG que es un modelo de ajuste fino basado en VGG para el reconocimiento de

carreras a partir, para la clasificación de la basura. Como resultado se vio la dificultad de clasificar la basura orgánica e inorgánica debido a las características similares de esta, por lo que en futuras investigaciones quieren estudiar con un modelo profundo para la clasificación de esos escenarios de desequilibrio.

Estudios Referentes a Trabajos Sobre Redes Neuronales Artificiales

- Jana *et al.* (2021) usan las técnicas de aprendizaje automático y la visión por computadora para problemas relacionados a las frutas y vegetales en cuanto al estado de descomposición en el que se encuentran. La clasificación de imágenes de las bases de datos se da entre frescas y dañadas. Por resultados se dio la buena práctica del modelo CNN entre el conjunto de 4 grupos (1. Manzana, 2. Plátano, 3. Naranja, 4. Anteriores 3) que se tuvieron al final.
- Ismail Jabiullah *et al.* (2021) se centra en la detección de enfermedades en los tomates, lo que para ellos usan las redes neuronales convolucionales como ayuda en la clasificación dada en 9 tipos de enfermedades. Como resultado se obtuvo la validación y la precisión del modelo mayor al 90% lo que se podría informar que tiene un entrenamiento muy bueno.
- Hasan *et al.* (2021) usan el aprendizaje automático y la visión por computadora para la clasificación de la mayor cantidad de enfermedades que puede obtener una papa en un cultivo. Como resultado se usó los métodos ResNet-50, GoogLeNet y AlexNet para luego dejar una claridad que son el uso de estos se pudo tener un mejor entrenamiento y precisión del producto final.

- Rahat *et al.* (2021) presenta un estudio sobre la clasificación de insectos en el cultivo de mango utilizando técnicas de aprendizaje profundo. El estudio se enfoca en la creación de un conjunto de datos de insectos que afectan la producción de mango en la región asiática y en la aplicación de un método de aprendizaje profundo basado en la combinación de tres arquitecturas de redes convolucionales (VGG19, MobilNet y Xception) para mejorar el rendimiento de la clasificación de insectos.
- Kabir *et al.* (2021) describe un modelo de diagnóstico de enfermedades de plantas óptimo para múltiples plantas utilizando redes neuronales convolucionales. Se presenta los resultados experimentales del modelo en diferentes conjuntos de datos de enfermedades de plantas y muestra que el modelo propuesto supera a los enfoques existentes en términos de precisión y eficiencia. El modelo propuesto puede ser útil para los agricultores y expertos en agricultura para detectar y diagnosticar enfermedades de plantas de manera temprana y precisa.

Formulación del Problema

En esta sección, se explora como la inteligencia artificial puede revolucionar la clasificación de residuos orgánicos. Se analiza a través de artículos relacionados en el campo, como la tecnología puede ayudar a abordar desafíos medioambientales y generar soluciones sostenibles.

Urbáez (2015) comenta que la exposición de innovación tecnológica en los diversos temas que rescatan las diferentes acciones que como individuos están asociadas, es el primer

paso para la unión de los principios que conllevan los cambios que hoy en día se han llegado a lograr; dentro de los mayores asuntos se encuentran: energético, financiero, social, político, de valores, climática, ambiental, guerras entre otras.

Entonces, así Gilbert (1995) dice que la tecnología se muestra como un enfoque que se usa como proceso y producto donde al identificar algunas de las necesidades como por ejemplo medioambientales; se podría ver una transparencia de buenas prácticas investigativas. La muestra de los proyectos que son alcances propuestos para cubrir esas necesidades medioambientales, son los que dan más alternativas a un mundo sostenible, el cual, al mostrarla en el campo de inteligencia artificial de las ciencias de la computación como se nombra en Herrera (1981) genera tecnologías apropiadas.

GALIPIENSO *et al.* (2003) especifica las áreas en las que la inteligencia artificial ha presentado más acercamiento, una de ellas es los problemas de percepción (Visión y Habla), que dentro de los antecedentes muestra el reconocimiento de objetos, detección o apoyo en diagnóstico.

El reconocimiento de objetos se toma como la clasificación de alguna información de cosas del mundo físico, así como nombra Rojas *et al.* (2008) siendo una detección de objetos mediando imágenes. Las imágenes que son de la detección de materiales orgánicos los cuales son desechos que nadie los está utilizando en el momento y que se les puede dar un nuevo uso; se llegara a la idea de una clasificadora que llegando a reciclar estos materiales orgánicos se ayudaría al medio ambiente con un espacio más limpio. Al tener en cuenta el reciclaje como un medio de ayuda, la clasificadora podría fabricar nuevos productos.

Gonzalez *et al.* (2017) afirma que, en Colombia ha venido creciendo el desarrollo de nuevos productos que contienen tecnología con relación a lo agroindustrial, lo que se plantea

buscar nuevos lugares donde se pueda construir un proyecto consistente logrando alternativas de este objetivo; se encontró en el Departamento de Cundinamarca un municipio que cumplía con ese llamado el cual es Sopó.

Pregunta de investigación:

¿Cómo clasificar residuos orgánicos mediante inteligencia artificial?

Justificación

La Inteligencia Artificial dicho por Martínez Llamas (2018) como la capacidad que una maquina puede llegar a tener para razonar y pensar efectivamente como un humano. Como objetivo de necesidades de proveer el sistema de conocimiento de la maquina nace el Machine Learning (Aprendizaje Automático) y el Deep Learning (Aprendizaje Profundo).

El aprendizaje profundo utiliza conjuntos de datos para así guiar al modelo que está hallando, para que al final de proceso se presente una adecuada presentación del problema inicial. Mencionando los ejemplos de diferencias entre el aprendizaje profundo y el aprendizaje automático que da Ponti y Da Costa (2018) son dos el ejemplo 1: Clasificación de imágenes y el ejemplo 2: Detección de objetos.

Se toma la clasificación de imágenes como ayuda a las buenas prácticas de clasificar residuos, lo que la intervención humana ha llegado a entender para mostrar sistemas entrenados capaces de analizar imágenes de forma autónoma y con resultados a fines. Las imágenes deberían ser en su mayoría de alta resolución para tener un análisis más claro donde la clasificadora pueda entender bien que clase o tipo de material el que se está hablando, pero no

solo eso también entre más data se tenga más se reduce el porcentaje de un resultado negativo ante lo que de verdad es el material (Muñiz, 2018). Las tareas más complejas que se tienen pueden ser realizadas con el aprendizaje automático, donde el sistema aprende en función a los datos que va adquiriendo, donde por el apartado podemos informar que en un nivel nacional colombiano se tiene que una de las metas del Plan Nacional de Desarrollo 2014 – 2018 (“Todos por un Nuevo País”), consistía en incentivar y fortalecer el aprovechamiento de residuos sólidos en los municipios de Colombia, en especial en aquellos que tuvieran más de 500.000 habitantes, ya que, en 2013, tan solo el 17% de los residuos sólidos municipales fueron aprovechados (Santos, 2014). Por ende, el municipio de Sopo busca desarrollar un modelo de gestión de residuos sólidos con el cual contribuir al mejoramiento del impacto ambiental, social y económico de Cundinamarca y dar alcance a los indicadores propuestos en el Plan Nacional de Desarrollo.

En el momento de creación del algoritmo que va a adquirir la máquina para la generación de alternativas de procesamiento y aprovechamiento de residuos sólidos orgánicos permitirá impulsar la investigación científica en el ámbito del desarrollo sostenible del municipio de Sopó acorde al artículo de (Payoma et al., 2022).

Objetivos

Objetivo General

Estructurar un modelo de inteligencia artificial mediante el reconocimiento de imágenes que sea capaz de clasificar la reutilización de residuos orgánicos.

Objetivo Especifico

Recolectar el conjunto de datos necesarios para la implementación de un modelo de inteligencia artificial para la clasificación de residuos orgánicos.

Construir un modelo de clasificación de imágenes mediante una Red Neuronal Convolutiva para la clasificación de los residuos orgánicos.

Evaluar y analizar la precisión del modelo previamente entrenado con diferentes métricas de inteligencia artificial.

Metodología

La metodología del trabajo aquí presentado es similar a la nombrada en Ismail Jabiullah *et al.* (2021) que consiste en la implementación de una red neuronal convolucional profunda para la detección temprana de enfermedades en hojas de tomate. Se utilizó una base de datos de imágenes de hojas de tomate sanas y enfermas para entrenar la red neuronal y se evaluó su desempeño utilizando medidas de precisión, recall y F1 score. Además, se comparó el desempeño de la red neuronal con otros métodos de detección de enfermedades en hojas de tomate.

Kabir *et al.* (2021) también muestra su metodología con el uso de redes neuronales convolucionales (CNN) para el diagnóstico de enfermedades en múltiples plantas. Utilizando una base de datos de imágenes de hojas de plantas enfermas y sanas para entrenar y validar el modelo de CNN. Además, se aplicó una técnica de aumento de datos para aumentar la cantidad de imágenes disponibles para el entrenamiento. Se evaluó las diferentes arquitecturas de CNN y se selecciona la que mejor rendimiento tuvo en términos de precisión, sensibilidad y especificidad. Finalmente, valida el modelo utilizando un conjunto de datos de prueba independiente.

Ya que ambos presentan una organización por fases para cuando quieren nombrar el proceso que se realizara, entonces tomamos aquello para presentar la metodología aquí seguidamente mencionada.

Las fases aplicadas son 3, la primera correspondiente al procesamiento de información en la cual se realizará una búsqueda imágenes en bases de datos existentes como por ejemplo el repositorio de GitHub, base de datos de imágenes de TensorFlow, y carpetas indexadas en sitios web sobre trabajos relacionados. Se tomarán los registros fotográficos in situ de los residuos orgánicos, recopilados por la ruta selectora.

También, se clasificará las imágenes de la base de datos entre material orgánico reutilizable o de compostaje y se implementará el proceso de aumento de datos mediante rotaciones, translaciones y zoom para facilitar el proceso de detección del prototipo.

La segunda fase es la comprensión del proceso de aprendizaje automático del prototipo específicamente al entrenamiento que este recibirá, donde se revisa el código y ejecuta el mismo.

Se tiene en cuenta de emplear la herramienta Google Colaboratory para luego desarrollar un código de programación en lenguaje Python mediante el uso de paquetes de software como lo son: Tensorflow, Keras, Numpy y el estado de interfaz matplotlib.pyplot. Se realiza una búsqueda sistemática de referencias relacionada con la teoría de redes neuronales convoluciones e investigar las diferentes funciones de activación con aplicación a redes neuronales convolucionales.

Así mismo se selecciona la función de activación que presente el mejor ajuste para las condiciones del proyecto para luego entrenar mediante el uso de las imágenes recopiladas por la información primaria y secundaria el modelo construido.

Finalmente, la tercera fase corresponde al análisis comparativo de los resultados obtenidos donde es complementar el entrenamiento del modelo mediante el uso de métricas de inteligencia artificial de lo que más tarde se valida el modelo mediante el uso de TensorBoard que nos permita lograr los objetivos propuestos en el proyecto y de mas tarde obtener los indicadores o métricas correspondientes que nos permitan presentar los porcentajes de ajuste del modelo para evitar overfitting (Sobreajuste).

Por otro lado, el modelo crisp DM, un enfoque estándar para el desarrollo de sistemas de información basados en datos, consta de cinco fases, desde comprensión del problema hasta la evaluación del modelo. Sin embargo, su aplicación directa en proyectos que utilizan redes neuronales convolucionales (CNN) puede ser limitada debido a la naturaleza diferente de estos modelos de aprendizaje automático. Las CNN son altamente efectivas en el procesamiento de datos de imagen, ya que pueden identificar automáticamente características visuales complejas. A diferencia de crisp DM, que se enfoca en datos estructurados, las CNN trabajan con datos no estructurados y requieren un enfoque de preprocesamiento y modelado específico.

Si bien los conceptos generales de preprocesamiento de datos y evaluación de resultados del modelo crisp DM podría aplicarse en proyectos que involucren redes neuronales convolucionales, se requiere una comprensión más profunda de la arquitectura de la red y ajustes precisos en términos de capas, funciones de activación y otros parámetros. Además, la evaluación de estos modelos se basa en métricas específicas de visión por computadora, como precisión, sensibilidad y especificidad, en contraste con las métricas más tradicionales utilizadas en el modelo crisp DM.

Fundamentación Teórica

En el presente capítulo se representa los fundamentos teóricos que respaldan las tecnologías utilizadas en nuestro estudio. Ahora bien, abordaremos conceptos clave como el análisis de clasificación de residuos orgánicos, inteligencia artificial, aprendizaje automático y aprendizaje profundo. Nos adentraremos en el mundo de las redes neuronales artificiales incluyendo conceptos como RGB, capas de convolución, funciones de activación y más.

Además, se examina la técnica de aumento de imágenes y su importancia para mejorar la capacidad de generalización de nuestros modelos. También consideraremos los pesos de clase y las métricas utilizadas para evaluar los clasificadores de aprendizaje automático. A medida que se avanza, se exploran los desafíos asociados con el sobreajuste y como se pueden abordar mediante arquitecturas de redes neuronales convoluciones y técnicas como el Dropout.

Para tener en cuenta, se explora un poco las herramientas y librerías clave utilizadas, lo que nos permitirá dar vida a los conceptos y realizar el experimento práctico.

Análisis de Clasificación de Residuos Orgánicos

El residuo se refiere a las sustancias u objetos que resultan de actividades de producción y consumo y que no tienen valor económico.

Se consideran residuos orgánicos aquellos que provienen de actividades como la agricultura, la ganadería, los mataderos, los residuos forestales, los residuos domésticos, los lodos de depuradoras de aguas residuales, así como los residuos generados por las industrias agroalimentarias y relacionadas (cascaras, cascaras de huevo, etc.).

Se encuentran sustancias orgánicas simples con una fórmula definida directamente relacionada con su origen. Además, también se presentan sustancias orgánicas de mayor complejidad estructural, que se forman en muchas ocasiones durante los procesos de tratamiento de los residuos mediante la acción de microorganismos que actúan sobre ellos. Estos microorganismos descomponen los residuos y generan compuestos orgánicos más complejos como resultado de sus procesos metabólicos (Pedreño *et al.*, 1995).

La generación de residuos sólidos es un problema global, especialmente en las zonas urbanas. Los países en desarrollo enfrentan desafíos significativos debido a las grandes cantidades de residuos sólidos que generan, lo que provoca graves problemas de contaminación ambiental (Al-Khatib *et al.*, 2010).

El aumento en la generación de residuos sólidos afecta negativamente los servicios sanitarios, el suministro de agua, la gestión de residuos y la infraestructura de transporte en las ciudades. Estudios demuestran que la recolección, almacenamiento, transporte y disposición final de los residuos sólidos son problemas importantes en áreas urbanas, tanto en África como en otros países en desarrollo. Estos problemas se deben principalmente a la limitada economía de estas regiones, lo que dificulta la gestión adecuada de los residuos sólidos. El tamaño promedio de las familias, el número de habitaciones, los ingresos mensuales y el empleo también influyen en la generación y composición de los residuos sólidos (Porrás & González, 2016). Factores socioculturales, económicos, legales, políticos y ambientales,

junto con los recursos disponibles, afectan la gestión de los residuos sólidos en todos los países. Además, los cambios en los patrones de consumo de las personas y los avances tecnológicos han llevado a cambios en las cantidades y composición de los residuos sólidos. Estos aspectos varían entre países, regiones e incluso dentro de una misma ciudad, dependiendo de los hábitos de consumo de la población (Abdel- Shafy & Mansour, 2018).

La economía circular es un enfoque alternativo al modelo económico lineal “tomar, hacer, tirar” que prevalece en la actualidad. Mientras que el modelo lineal depende de la disponibilidad abundante y barata de materiales y energía, así como de opciones económicas para desechar lo que ya no se necesita, este enfoque está llegando a sus límites físicos y no es sostenible a largo plazo.

La economía circular se basa en la idea de mantener los productos y los materiales en uso durante el mayor tiempo posible, promoviendo la regeneración y el aprovechamiento máximo de los recursos disponibles. Este enfoque busca superar las limitaciones del modelo económico lineal y abordar los desafíos ambientales y de sostenibilidad asociados con él (Cerdá & Khalilova, 2016).

En Colombia se ha generado nuevas políticas para gestionar los residuos orgánicos como normativa ambiental. El compostaje es una técnica ampliamente utilizada en Colombia para aprovechar los residuos sólidos orgánicos urbanos. Consiste en descomponer los residuos orgánicos mediante la acción microbiana, lo que cambia su estructura molecular. Dependiendo del tiempo de degradación, se determina el grado de madurez del compost, que puede ser parcial (descomposición de un compuesto orgánico en otro similar) o completa (descomposición total en dióxido de carbono). Esta opción es ampliamente preferida debido a

su capacidad para tratar grandes cantidades de residuos, como en el caso de la generación de residuos sólidos urbanos (Vargas *et al.*, 2011).

Composición de los Residuos Orgánicos

A lo largo de la historia, las actividades humanas han llevado a un aumento en la generación de residuos o materiales descartados, especialmente en las zonas urbanas, con una proporción significativa de residuos sólidos orgánicos. Según Pedreño *et al.* (1995), estos residuos orgánicos presentan una composición principalmente por materiales de origen vegetal y animal, como, por ejemplo, restos de alimentos, poda de plantas, estiércol, residuos de cultivos, entre otros. Estos materiales contienen una gran cantidad de nutrientes, como nitrógeno, fósforo y potasio, que son esenciales para el crecimiento de las plantas; así mismo al ser de origen natural, presentan propiedades para desintegrarse rápidamente (Biodegradables).

Tratamiento de los Residuos Orgánicos

Existen diferentes métodos de tratamiento de los residuos orgánicos, entre los cuales se destaca el compostaje. Para Villegas-Cornelio y Laines Canepa (2017), el compostaje es un proceso natural de descomposición de la materia orgánica, en el cual los microorganismos descomponedores transforman los residuos en un producto llamado compost. Este compostaje se puede realizar a diferentes escalas, como en los hogares o comunidades; o más ampliamente en instalaciones municipales de compostaje. Por otro lado, existe otro proceso de tratamiento de residuos orgánicos (Vermicompostaje), que según Villegas-Cornelio y Laines Canepa (2017), se

utilizan lombrices y microorganismos para la biooxidación o descomposición de los residuos. Este proceso es muy eficiente y económico. Así, estos y otros métodos pueden reducir la cantidad de desechos que van a los vertederos, limitar los gases de efecto invernadero y crear un producto nutritivo para usar en la agricultura.

Aplicaciones del Compost

El compost obtenido a partir de los residuos orgánicos tiene diversas aplicaciones. Para el PROGRAMA SIRSD-S, en la agricultura, se utiliza como fertilizante orgánico, enriqueciendo el suelo con nutrientes y mejorando su estructura, esto puede aplicarse en diversos cultivos y terrenos. Así mismo, se puede utilizar en jardinería, para mejorar la calidad del suelo y promover el crecimiento de las plantas; eso sí, teniendo en cuenta que la incorporación del compost debe realizarse en suelos adecuadamente húmedos y antes de la siembra. Además, el uso de los residuos orgánicos en forma de compost contribuye a la reducción de la contaminación ambiental, ya que logra reducir los gases de efecto invernadero y la generación de lixiviados en los vertederos donde se depositan la mayoría de estos residuos (PÉREZ, 2014).

Según Galdames (2016), se ha encontrado una aplicación muy útil en la agricultura orgánica: el uso de un té de compost o un extracto acuoso obtenido mediante un biodigestor. Este proceso implica la combinación de compost de alta calidad, un líquido extractor y algunos aditivos para estimular el crecimiento de microorganismos beneficiosos, como energía y proteínas. Esta alternativa resulta realmente valiosa para la agricultura orgánica, ya que contribuye a mejorar la salud del suelo y promueve el desarrollo de microorganismos beneficiosos.

Por otro lado, mencionando lo que dice Kashmanian y Rynk (1996), en diferentes países alrededor del mundo están implementando regulaciones y aprovechamiento de los residuos orgánicos. Por ejemplo, en los Países Bajos, el compostaje se ha convertido en una parte fundamental de su estrategia para gestionar los residuos, ayudando a reducir la cantidad de desechos y fomentando la producción de energía renovable a partir de ellos. En Estados Unidos se han puesto en marcha programas obligatorios para el compostaje, logrando desviar grandes cantidades de residuos que iban a parar a los vertederos y contribuyendo así a reducir las emisiones de gases contaminantes. Además, en Nueva Zelanda se ha integrado el compostaje dentro de las prácticas agrícolas sostenibles con resultados muy positivos: mejora la calidad del suelo y reduce la necesidad del uso excesivo de fertilizantes químicos.

Teniendo en cuenta esto, el uso del compost puede ayudar a restaurar ecosistemas que han sido degradados, además de promover la participación ciudadana y la conciencia ambiental. Esto contribuye a la sostenibilidad y también ayuda a abordar el creciente desafío de garantizar la seguridad alimentaria.

Inteligencia Artificial

Rouhiainen (2018) brinda una definición de inteligencia artificial detallada diciendo que la inteligencia artificial muestra una competencia de las máquinas en el uso de los algoritmos que aprenden de los datos y el uso de la toma de decisiones, realizado de la misma manera que lo haría un humano. Tal y como se menciona, la Inteligencia Artificial (IA) posee dos metas primordiales. La primera radica en el ámbito tecnológico, empleando computadoras para llevar a cabo tareas prácticas, a menudo utilizando enfoques distintos a los procesos mentales. La

segunda se enfoca en el ámbito científico, utilizando conceptos y modelos de IA para abordar cuestiones relacionadas con los seres humanos y otras formas de vida. Aunque la mayoría de los expertos en IA se enfocan en una de estas metas, algunos consideran ambos objetivos (Boden, 2017).

Aprendizaje Automático

Por medio de El Naqa y Murphy (2015) decimos que el aprendizaje automático es una rama en evolución de los algoritmos computacionales que están diseñados para emular la inteligencia humana aprendiendo del entorno circundante. Hoy en día ha sido puesto en área de agricultura, medicina, economía, bioinformática entre otras las cuales ha llevado a solventar los problemas que antes no se podían como la detección de carros robados en una ciudad por medio de la toma de imágenes de las placas, o el uso para la predicción y toma de decisiones en el cambio climático ante la afectación de la producción agronómica.

Figura 1

Proceso de análisis de datos en el aprendizaje automático



Nota. Las fases están basadas en el estándar CRISP-DM (ACOSTA-GUZMAN et al.).

En cuanto a los tipos de aprendizaje automático Rouhiainen (2018) expone que se obtienen tres: El primero llamado aprendizaje supervisado que está basado en tareas, aprendizaje no supervisado el cual es basado en datos y por último el aprendizaje de refuerzo donde aprende a reaccionar a su entorno. Por medio de ello el proyecto se enfoca en el aprendizaje supervisado, dado los algoritmos y el uso de los datos que ya se han organizado para luego crear unas categorías de cómo se podría visualizar o caracterizar la información, de esta manera se puede tener en cuenta que si se habla de materiales orgánicos por todas sus etapas desde estar madura hasta su estado biodegradable con el que al momento de concentrarnos en ella se pueda a través de identificación de imágenes saber que se habla de manzanas.

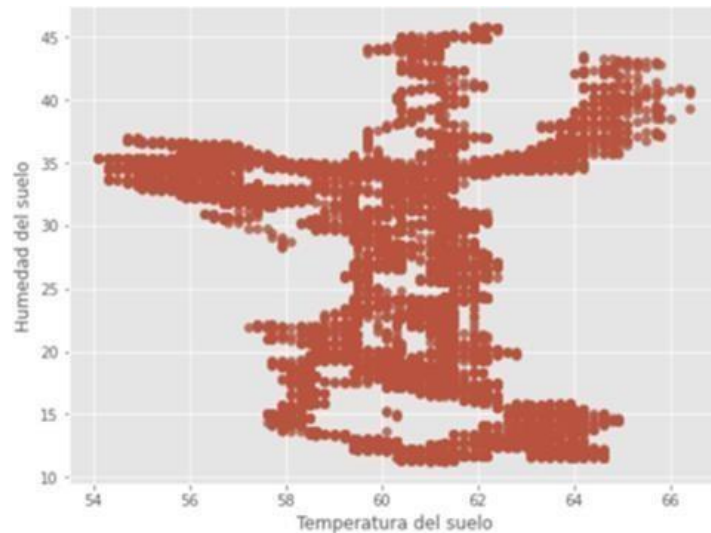
Aprendizaje Supervisado

Inicialmente inicia con la información recolectada recordando que esa información son datos los cuales más adelante se les dará un entrenamiento y prueba para luego ser clasificados o de predicción. Bonaccorso (2017) nombra que la meta es entrenar un sistema que pueda también trabajar con muestras nunca vistas antes, lo que es necesariamente dejar al modelo que desarrolle una habilidad generalizaciones y evite un problema llamado sobreajuste.

Dentro de los que tenemos como aprendizaje supervisado son trabajos como el análisis predictivo basado en regresión figura 2, la detección, clasificación automática de imágenes el procesamiento secuencial automático entre otros.

Figura 2

Vista de una de las correlaciones en este caso las variables Humedad del suelo y Temperatura.



Fuente: Autor

Aprendizaje no Supervisado

Es usado mayormente cuando nos referimos a temas de clustered (Agrupados), en el cual nos ayuda a asimilar las medidas e intervalos que se manejan, como por ejemplo mirar figura 2, puedes observar y detectar que colores fueron usados para la realización de esta y llegar a describirlos. Los algoritmos de aprendizaje no supervisados aprenden pocas características de los datos. Cuando se introducen nuevos datos, utiliza las características previamente aprendidas para reconocer la clase de los datos (Mahesh, 2020).

Aprendizaje Profundo

Cuando tenemos en cuenta un aprendizaje automático podemos hablar de un mayor crecimiento de inteligencia artificial al referirnos al campo de Deep learning (aprendizaje profundo), el que nos ayuda a resolver problemas más complicados y a un problema que tengan mayor cantidad de datos por procesar. Como nombra LeCun *et al.* (2015) los métodos de aprendizaje profundo son método de aprendizaje de representación multinivel, obtenidas mediante la combinación de módulos simples, pero no lineales, esto queriendo decir que se buscan los aspectos más relevantes y discriminatorios desde la entrada de los datos para realizar las tareas de clasificación superior.

Redes Neuronales Artificiales

Una red neuronal es como un conjunto de pasos que trata de entender las conexiones ocultas en un montón de información, imitando cómo funciona el cerebro humano. Básicamente, es como tener un sistema de neuronas, ya sea en un cerebro de verdad o en una computadora. Lo sorprendente es que estas redes pueden ajustarse a los cambios que recibe, de manera que puedan ofrecer la mejor respuesta posible sin necesidad de rehacer todo el proceso (Wei et al., 2019).

Se menciona que hay dos tipos de redes neuronales artificiales (ANN): La red neuronal recurrente (RNN) y la red neuronal convolucional (CNN), se explica que la RNN es adecuada para el análisis de series de tiempo debido a su capacidad para procesar datos secuenciales y su capacidad para recordar información previa. Además, la RNN tiene una estructura de bucle que permite que la información se retroalimentación la red, lo que le permite recordar información previa y utilizarla para hacer predicciones más precisas. Por otro lado, la CNN es adecuada para

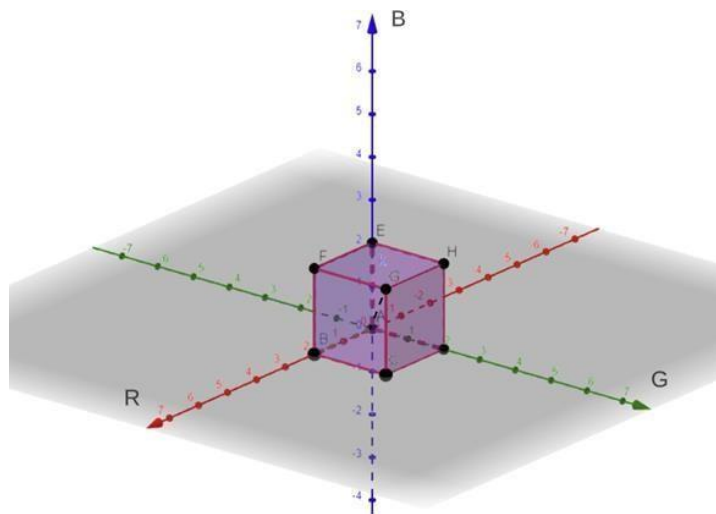
el procesamiento de imágenes porque utiliza la convolución para extraer características de las imágenes de entrada.

RGB

Por sus siglas en inglés RGB (Red, Green, Blue) es un proceso de creación de colores basado en tres señales distintas de luminancia cromática¹. Para obtener un color deseado, se ajustan las cantidades de estos tres colores. Gráficamente, esta combinación aritmética de los valores X de estos colores se representa en un cubo figura 3, que conecta dos de sus vértices (Blanco y negro), representa la escala de grises. En esta escala, podemos encontrar distintos tonos de gris entre estos dos extremos (Pérez, 2009).

Figura 3

Grafica de cubo RGB realizada en GeoGebra

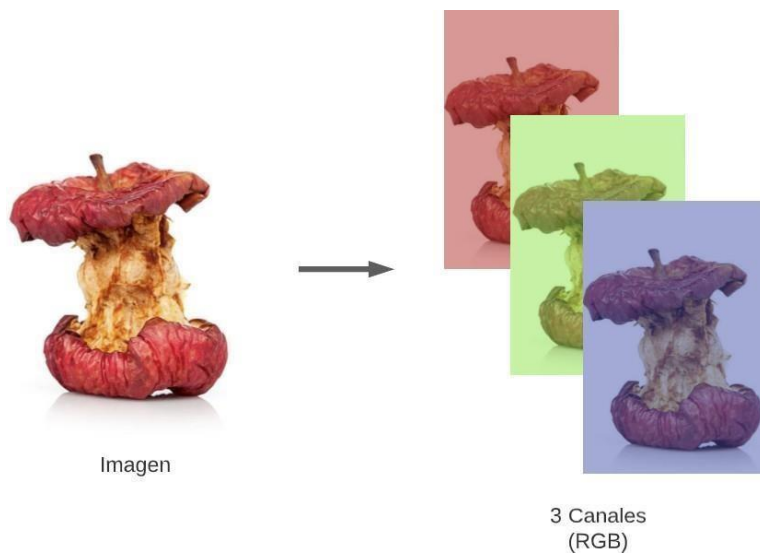


¹ La luminancia cromática, también conocida como contraste cromático, se refiere a las variaciones en la composición espectral de una imagen y, por ende, en los matices percibidos por un observador humano (Lloreda, 2001). Fuente: Autor.

Para cuando tenemos una imagen detectamos lo que significa el espacio de color comenzando por unidimensional que en causa tiene la pérdida espacial de píxeles, el bidimensional donde al hacer un proceso extrae varias características con variables en dos ejes x , y y el tridimensional que llega a ser visto como el espacio de color RGB ya que cada píxel tiene los tres componentes o canales de colores básicos (Rojo, verde y azul) visto en la figura 4.

Figura 4

Imagen desde una perspectiva de espacio de color RGB



Fuente: Autor

Convolución

Si se toma una imagen a escala de grises como en la figura 5, se representa cada píxel² con valores enteros dentro de un rango de $[0,255]$, donde si la tonalidad es oscura se dan valores que se aproximen al 0, si son de variables intermedias se toma entre 128 y para las tonalidades claras o brillosas números dados cerca al 255.

Figura 5

Imagen en escala de grises y la muestra de una fracción de la imagen como pixel



Fuente: Autor

En las redes neuronales convolucionales, un filtro se refiere a una matriz de parámetros utilizada para realizar la operación de convolución en los datos de entrada. También conocido como Kernel³, en la mayoría de los casos observados con líneas verticales para así poder buscar ágilmente patrones en la imagen original como se muestra en el Tabla 1.

Al tomar un kernel y la foto original se mostrará totalmente diferente a lo que ya antes hemos visto en la figura 5, lo que a partir de ese resultado se muestra una imagen diferente,

² Pixel es la unidad básica de una imagen digitalizada en pantalla a base de puntos de color o en escala de grises.

³ Kernel es el desplazamiento hacia la derecha o debajo de un píxel durante cada iteración.

entonces podemos llamar a la operación un kernel y una imagen una convolución que ayuda a detectar patrones importantes para la investigación.

Tabla 1

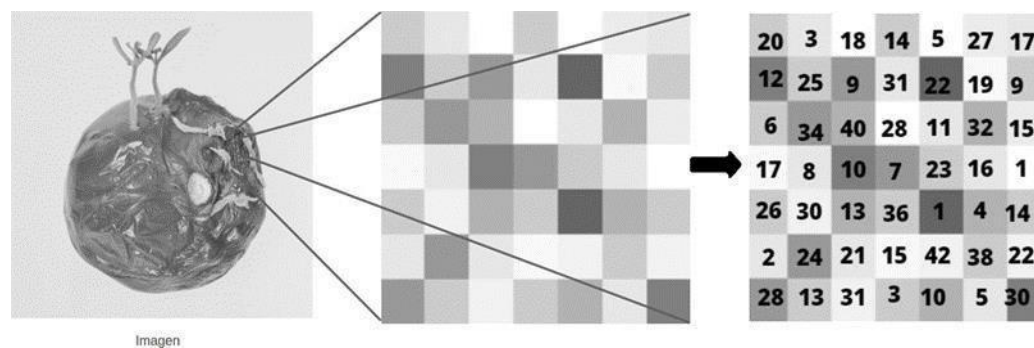
Un filtro de tamaño 3×3

1	1	1
-1	-1	-1
0	0	0

Fuente: Autor

Figura 6

Nueva imagen por medio de la original dado el concepto de kernel



Fuente: Autor

Se aplica de dos o más formas, la primera se presenta mediante dos dimensiones y por medio de una segunda puede ser dada en tres dimensiones esto simplemente dado como función de las imágenes que se interpretan.

La función bidimensional está dada en dos variables que se vuelven coordenadas especiales. La función está dada de la forma $z = F(x, y)$.

Capas de una Red Neuronal

Las capas de una red neuronal artificial como informa Acevedo *et al.* (2017), son las conectoras de información a través de las conexiones que ellas mismas hacen. La capa de entrada llega a ser la que primero recibe la información y de ese modo la capa de salida es la que muestra lo que resulto luego de hacer una transformación a la información; pero también se encuentran unas capas llamadas ocultas siendo estas las que realizan el procesamiento y trabajo de lo que la capa de entrada recibió.

Las capas entonces llegan a ser unidades de procesamiento de datos. Dentro de la arquitectura de la red neuronal podemos encontrar las capas.

Capa de Convolución

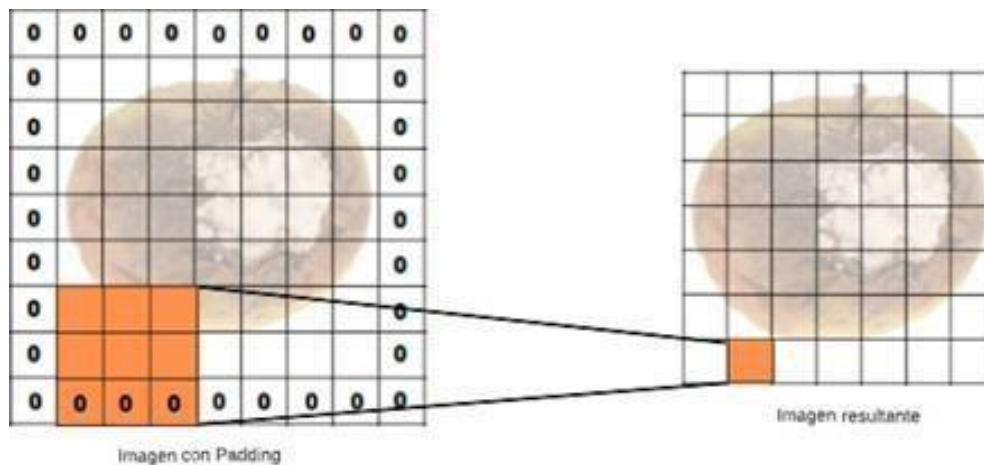
Recordando que los filtros o kernel es una matriz bidimensional que se desliza sobre los datos de entrada mediante la operación de convolución, podemos decir que la capa de convolución está compuesta por múltiples filtros que se aplica a los datos de entrada y cada filtro realiza la operación de convolución.

La operación de convolución se especifica mediante parámetros como el stride (paso), el tamaño del filtro y el zero padding (relleno con ceros). El stride determina el tamaño del paso del filtro mientras se desliza sobre la entrada. El tamaño del filtro, también conocido como campo receptivo, se mantiene fijo en todos los filtros utilizados en la operación de convolución (Albawi *et al.*, 2017).

El padding es una técnica que implica agregar píxeles con valor cero en los bordes de una imagen original. Esto permite que la imagen resultante tenga el mismo tamaño que la original (Duran et al., 2021).

Figura 7

Capa de relleno de ceros añadidos a la imagen original



Fuente: Autor

Al aplicar capas de convolución con diferentes filtros y configuraciones, una CNN puede aprender representaciones jerárquicas de los datos de entrada, capturando características importantes a diferentes escalas y ubicaciones espaciales. Esto hace que las capas de convolución sean eficaces para tareas como reconocimiento de imágenes y visión por computadora.

Capa de Pooling

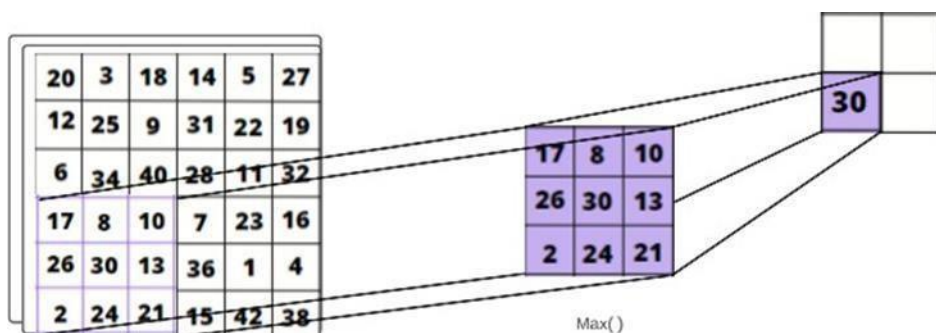
La capa de pooling o de agrupación normalmente actúa como la convolución de dos capas que se encuentran de salida de las convoluciones que se están haciendo. El objetivo de las

capas de agrupación es reducir gradualmente la dimensionalidad de los datos, lo que a su vez disminuye el número de parámetros y la complejidad del modelo. Esto ayuda a controlar el problema de sobreajuste y facilita el procesamiento de los datos de manera más eficiente (Sakib et al., 2019).

Proceso de Submuestreo

Sanchez-Guashpa *et al.* (2021) indica que el proceso dado como submuestreo permite optimizar el rendimiento computacional para reducir el tamaño del espacio en el aprendizaje de nuevos datos dados estos en cualquier lugar de la imagen que se tiene. Dentro de los más reconocidos es Max Pooling reduciendo la cantidad de patrones a aprenderse y busca características detalladas de ayuda; también podemos ver otros procesos como average pooling y sum pooling.

Figura 8
Ilustración de Max Pooling



Fuente: Autor

En la técnica Max Pooling, se divide la entrada en regiones no superpuestas y se selecciona el valor máximo dentro de cada región y se descartan los demás valores. Esto resalta las características más destacadas y reduce el tamaño de los datos (Shyam, 2021).

Funciones de Activación

Proveniente de la matemática elemental se encuentran unas funciones que como se dice en Nacelle y Mizraji (2009) son deterministas dando una monotonía en cuanto a la posición que se muestran en grafica como lo es ser creciente y continua. Comentado por López y Fernández (2008) las funciones más empleadas están dadas de la siguiente manera:

- Identidad

$$y = x$$

Es empleada en redes de baja complejidad siendo aquella que simplemente retorna el valor de activación de la neurona sin realizar ninguna transformación adicional.

- Escalón

$$y = \text{sign}(x)$$

$$y = H(x)$$

La función escalón es una función que representa salidas binarias, generalmente tomando valores 0,1 o $-1,1$. En este caso, si la activación de la neurona es menor que un umbral determinado, la salida se asocia con un cierto resultado, mientras que, si es igual o mayor al umbral, se asocia con el otro resultado. Aunque las

neuronas definidas por esta función son fáciles de implementar, sus aplicaciones se limitan a problemas binarios.

- Lineal a tramos

$$f(x) = \begin{cases} -1, & \text{si } x < -1 \\ x, & \text{si } -1 \leq x \leq 1 \\ +1, & \text{si } x > 1 \end{cases}$$

Combina la simplicidad de la función escalón con una adaptación suave a través de la función lineal en una zona intermedia. Esta aproximación puede ofrecer una representación más realista de las respuestas neuronales y es fácil de calcular, lo que la hace más útil en diversas aplicaciones.

- Sigmoidea

$$y = \frac{1}{1 + e^{-x}}$$

$$y = \text{tgh}(x)$$

Es una especie de curva que tiene unos límites arriba y bajo y siempre va en una dirección. Lo interesante de estas curvas sigmoideas es que su pendiente nunca es negativa, y en sus extremos, cuando miras muy lejos, la pendiente es cero. En el centro, cuando el valor es cero, la pendiente es más empinada.

- Gaussiana

$$y = Ae^{-Bx^2}$$

La función tiene forma de campana de Gauss y puede adaptarse al mover su centro, ancho y altura, lo que las hace flexibles. Suelen usarse en redes complejas con múltiples capas ocultas, y se aplican reglas de propagación basadas en calcular distancias cuadráticas entre entradas y pesos de la red.

- Sinusoidal

$$y = A \sin(\omega x + \varphi)$$

Esta produce resultados ininterrumpidos dentro del rango de $[-1, +1]$.

Usualmente, se utilizan en situaciones donde se necesita una repetición de patrones en el tiempo de manera evidente.

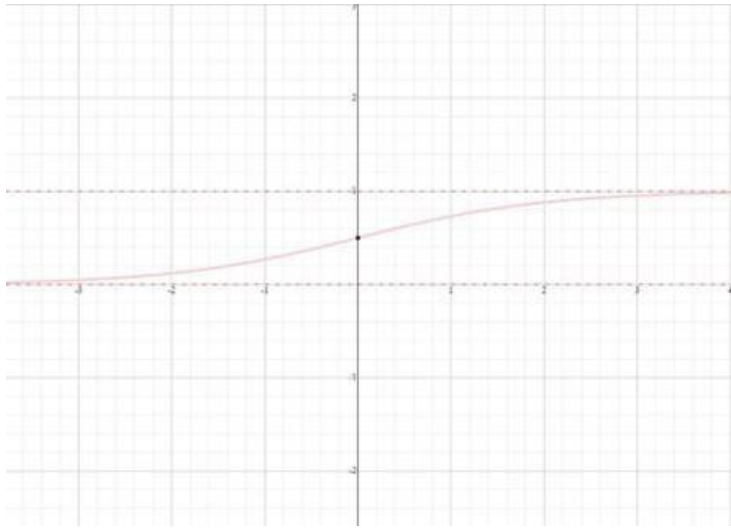
Estas funciones se escogen dependiendo del adaptamiento que se le quiere dar al modelo para que procese una cantidad de datos y que luego pueda hacerse predicciones precisas. Para resolver un problema de clasificación de base de datos implementada en redes neuronales convolucionales se tienen en cuenta las funciones de activación: Sigmoid, Softmax y RELU.

Sigmoid

La función es no lineal y su rango es de 0 a 1, con forma similar a una “S”. Es sensible a pequeños cambios en la entrada cercana a 0, pero tiende a producir salidas cercanas a 1 en los extremos. Esto puede dificultar el aprendizaje debido a los “gradientes de fuga”. Aun así, se utiliza en clasificación binaria, donde el resultado es 0 o 1. Si el valor es mayor a 0.5, se predice 1; de lo contrario, se predice 0 (Karlik & Olgac, 2011).

Figura 9

Grafica de la función de activación Sigmoid en el plano 2D.



Nota: Referencia tomada de Feng y Lu (2019).

ReLU

La función de activación ReLU se emplea en cada operación de convolución para realiza un cálculo sencillo que mantiene el valor de entrada sin alteraciones. Cuando la función recibe una entrada no positiva, devuelve cero, mientras que, para valores positivos de x, retorna el mismo valor (Kabir *et al.*, 2021).

$$f(x) = \max(0, x)$$

SoftMax

La función softmax es una extensión de la función sigmoidea que se puede utilizar para problemas de clasificación multiclase. Mientras que la función sigmoidea es adecuada para la

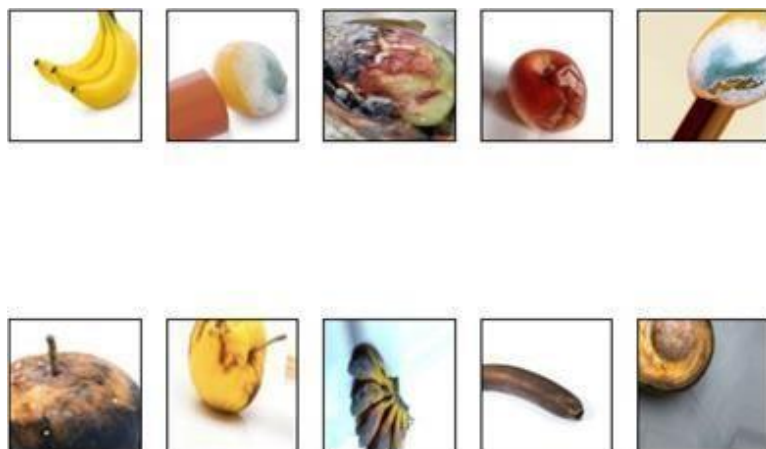
clasificación binaria, la función softmax nos permite asignar probabilidades a varias clases. Cada clase está asociada con una neurona en la capa de salida de la red, y la función softmax calcula la probabilidad de cada clase para un punto de datos determinado. En esencia, la función softmax combina múltiples funciones sigmoideas para proporcionar una distribución de probabilidad sobre las clases. En una red diseñada para clasificación multiclase, el número de neuronas en la capa de salida corresponde al número de clases en la variable objetivo (Sharma *et al.*, 2017).

Aumento de Imágenes

La técnica de aumento de imágenes se utiliza para incrementar la cantidad de imágenes en un conjunto de datos. Se aplican diversas transformaciones a todas las muestras, incluyendo rotaciones en cinco direcciones distintas (15° , 30° , 45° , 60° , 75°), así como translaciones y volteos verticales (Jana *et al.*, 2021).

Figura 10

Aumento de datos de las imágenes de residuos orgánicos clasificados entre reutilizable y de compostaje



Fuente: Autor

Como comenta Biswas *et al.* (2023) algunos de los enfoques mencionados son: Voltear, rotación, escalado, translación, recortar, aumento de color, mezcla y Recorte triangular. Estas técnicas son ampliamente utilizadas en campos como la visión por computadora y el aprendizaje automático, y han demostrado ser efectivas para mejorar la resistencia y capacidad de generalización de los modelos.

Se debe recordar que el aumento de datos es una estrategia clave en el análisis de imágenes para fortalecer y mejorar la capacidad de generalización de los modelos. Los datos son un componente fundamental en las aplicaciones de inteligencia artificial. Cuando hay escasez de datos etiquetados, es común que los modelos sufran de sobreajuste, lo que limita su capacidad de generalización a nuevas muestras. El aumento de datos aborda este problema al incrementar de manera efectiva tanto el volumen como la diversidad de los datos que son procesados por la red.

Pesos de Clase

Son una técnica utilizada para abordar desequilibrios en los conjuntos de datos de entrenamiento. En muchas aplicaciones de visión por computadora, es común que algunas clases o categorías estén representadas por un número mucho mayor o menor de ejemplos en comparación con otras clases (Velardo & Dugelay, 2012).

Class Weight for Imbalanced Dataset

El peso de clase en un conjunto de datos no balanceado es un método utilizado para corregir el costo de los errores de clasificación para diferentes clases en un conjunto de datos. Este conjunto de datos desequilibrado, donde una clase tiene significativamente menos ocurrencias que la otra, los clasificadores tradicionales tienden a clasificar todos los datos en la clase mayoritaria, que suelen ser la clase menos importante. La idea básica detrás del peso de clase es compensar los desequilibrios en la muestra de entrenamiento sin cambiar la distribución de la clase (Kotsiantis *et al.*, 2006).

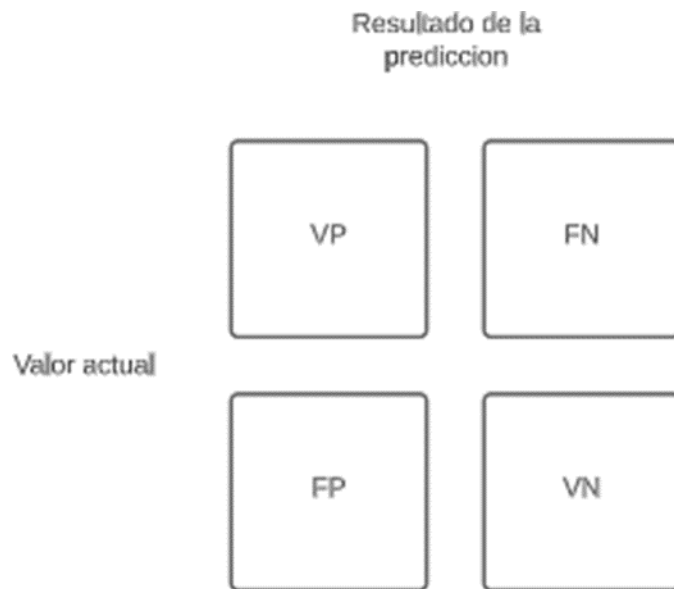
Métricas para Clasificadores de Aprendizaje Automático

Las métricas de clasificación se pueden clasificar en tres grupos principales: binarias, multiclase y multi etiqueta. Los clasificadores binarios se utilizan para predecir si una muestra pertenece a una de dos clases posibles, como responder preguntas de sí o no (Erickson & Kitamura, 2021).

Las métricas de evaluación son seleccionadas de acuerdo con la tarea en particular que se esté realizando.

Figura 11

Matriz de confusión. Nota. VP = Positivo verdadero, FN = Falso Negativo, FP = Falso Positivo, VN = Verdadero Negativo



Fuente: Autor

Algunas de ellas llegan a ser las más comunes como:

1.

$$Precision = \frac{VP}{VP + FP}$$

2.

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

3.

$$F1 - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Sencibilidad}}$$

4.

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

En la evaluación del rendimiento de la clasificación, la Exactitud o accuracy es comúnmente utilizada como la métrica principal. Se calcula periódicamente durante el proceso de evaluación y representa el porcentaje de muestras clasificadas correctamente, por eso fue la elegida para este trabajo (Mora *et al.*, 2021).

Sobreajuste

El sobreajuste ocurre cuando una red neuronal no puede aprender de manera óptima debido a varias razones. Es un concepto fundamental en la mayoría, si no en todos, los algoritmos de aprendizaje automático. Es importante tomar todas las precauciones necesarias para mitigar sus efectos, tanto en el conjunto de datos de entrenamiento como en los conjuntos de prueba y predicción (O'Shea & Nash, 2015).

Técnicas de Dropout

El método Dropout, como menciona Antona Castañares (2020) en su tesis de maestría, esta técnica de regularización se basa principalmente en la eliminación de neuronas en las capas de la red neuronal, teniendo una aplicabilidad determinada por una distribución Bernoulli; la técnica busca mitigar el sobreajuste (Overfitting), el cual se ajusta demasiado a los datos de entrenamiento y se vuelve demasiado especializado para esos datos específicos, perdiendo la capacidad de generalizar a datos nuevos o no vistos antes (Antona Castañares, 2020).

```
tf.keras.layers.Dropout()
```

En la capa Dropout se desactivan aleatoriamente algunas unidades de entrada en 0 durante el entrenamiento con una frecuencia *rate* en cada paso durante el entrenamiento, para evitar el sobreajuste; las unidades no desactivadas se escalan en $1/(1 - rate)$ para mantener la suma constante. Este solo se aplica cuando *training* es *True*, y no se eliminan valores de la inferencia. Finalmente, la configuración *trainable = False* no afecta el comportamiento, ya que Dropout no tiene variable que se congelen durante el proceso de entrenamiento.

Arquitecturas CNN

Como habla Chanampe *et al.* (2019) el uso de técnicas del aprendizaje profundo es de ayuda para la clasificación de imágenes en diferentes aspectos, lo mismo el destacar las CNN como método. También habla sobre las arquitecturas de Redes Neuronales convolucionales y su uso en el análisis de imágenes, así como su capacidad para descubrir características en las imágenes, desde la detección de bordes y esquinas hasta la detección de características más complejas a través del entrenamiento con grandes conjuntos de datos.

MobileNetV2

Es una versión mejorada y optimizada de MobileNet original, una arquitectura de red neuronal convolucional (CNN) diseñada para aplicaciones de visión artificial en dispositivos móviles y con recursos limitados. El principal avance de MobileNetV2 es el uso de bloques

residuales inversos con funciones de cuello de botella que son flexibles en términos de tamaños de entrada. A diferencia de algunas arquitecturas que requieren imágenes de tamaño fijo, MobilNetV2 admite todos los tamaños de entrada superiores a 32×32 píxeles. Esto permite que el modelo se adapte a diferentes tareas y requisitos, y también permite el uso de tamaños de imágenes más grandes, lo que a menudo resulta en un mejor rendimiento y precisión para las tareas de visión artificial (Sandler *et al.*, 2018).

MobileNetV3Large

Es una arquitectura de red diseñada específicamente para teléfonos móviles y dispositivos con recursos limitados. Esta arquitectura se desarrolló utilizando la técnica de búsqueda de arquitectura automatizada (NAS) con el algoritmo NetAdapt⁴ y otras mejoras de diseño.

El proceso de desarrollo de MobileNetV3 crea dos modelos diferentes: MobileNetV3-Large y MobileNetV3-Small, diseños para usar con recurso grandes y pequeños. Estos modelos han sido adaptados y aplicados a tareas como detección de objetos y la segmentación semántica (predicción de píxeles densos) (Howard *et al.*, 2019).

VGG16

Consta de un total de 16 capas, incluidas 13 capas convolucionales y 3 capas completamente conectadas. Las capas convoluciones consisten en filtros convolucionales que extraen

⁴ NetAdapt es un algoritmo diseñado para adaptar automáticamente una red neuronal preentrenada para su ejecución en plataformas móviles, teniendo en cuenta un presupuesto de recursos específico (Yang *et al.*, 2018).

características importantes de las imágenes de entrada. Estas capas se dan como bloques, cada una con múltiples filtros, y a medida que avanzan, el tamaño de las imágenes disminuye mientras que aumenta la cantidad de canales de las funciones aprendidas. Es conocido por usar pequeños filtros convolucionales (convolución 3x3) y capas convolucionales repetidas, lo que lo hace más profundo en comparación con otras arquitecturas populares como LeNet o AlexNet (Simonyan & Zisserman, 2014).

VGG19

Consta de 19 capas, incluidas convoluciones, capas de agrupación y una capa totalmente conectada para la clasificación. Su enfoque en las convoluciones de 3×3 y la uniformidad del diseño permitieron el aprendizaje de representaciones visuales significativas, lo que llevó a un rendimiento superior en las tareas de clasificación de imágenes. VGG19 es un modelo preentrenado ampliamente utilizando y sus pesos están disponibles públicamente en ImageNet (Simonyan & Zisserman, 2014).

ResNet50V2

Esta arquitectura particular tiene 50 capas y es una versión mejorada de la red ResNet50 original. La idea principal detrás de las redes ResNet es superar la degradación del rendimiento a medida que aumenta la profundidad de la red. Si agregamos, as capas en la red neuronal profunda, puede ser difícil entrenar el modelo y mejorar el rendimiento. Esto se debe a que las capas profundas pueden tener dificultades para aprender y pasar información importante de las capas originales a las capas posteriores (He *et al.*, 2016).

ResNet101V2

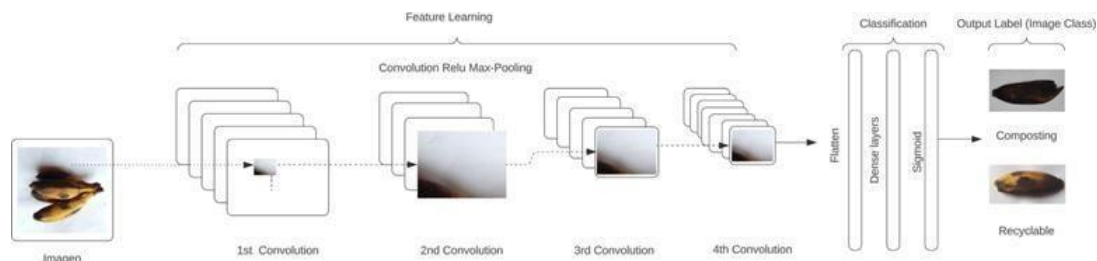
Es una modificación de la red neuronal convolucional ResNet (Restricción), una mejora de la arquitectura ResNet101 original. Utiliza 101 capas, la idea principal detrás de la arquitectura ResNet (Red Residual) es usar bloques residuales, lo que permite que la red aprenda atajos o residuales para saltar capas. Esto ayudara a suavizar el degradado, un problema que puede ocurrir en redes muy profundas. El desvanecimiento de gradiente puede dificultar el entrenamiento de redes neuronales muy profundas porque la información de las capas más profundas se pierde, lo que hace que la red se detenga o degrade su rendimiento (He *et al.*, 2016).

Redes Neuronales Convoluciones Profundas

Martínez Llamas (2018) comenta que las redes neuronales convolucionales son un tipo de redes neuronales con múltiples capas las cuales proveen un mejor reconocimiento y clasificación de las imágenes dadas. Lo que un computador se convierte en un humano al percibir las imágenes, pero de una manera diferente en el proceso. Las redes neuronales vienen dadas en capas las cuales son: entrada, ocultas y salida, donde cada capa cumple una funcionalidad esencial para el manejo de todo el sistema. Mediante las capas ocultas y las capas de entrada tenemos la diferencia de tener una red neuronal normal a una convolucional. Las CNN surgen mediante la necesidad de clasificar imágenes como imágenes de residuos orgánicos para reciclaje o compostaje. Se presentan entonces que las CNN son una solución a un problema basándose en la presentación y función de las conexiones que se dan con el cerebro y los ojos humanos.

Figura 12

Se representa el esquema que conlleva una imagen en la red neuronal convolucional, la cual pasa por las capas de entrada, ocultas y salida, donde en la salida ya sale con la clasificación de imagen, ya sea reutilizable o de compostaje.



Fuente: Autor

Herramientas y Librerías

Python

Es un lenguaje de programación que ha venido tomando poder por su alta velocidad de ejecución, sintaxis clara y la cantidad de librerías disponibles que maneja. Gonzales Duque (2011) informa que Python está disponible como herramienta multiplataforma lo que hace que corra sin problemas por los sistemas más comunes que se usan.

Para instalar Python, simplemente descarga el instalador que corresponde a tu sistema operativo desde la página oficial de Python. Una vez que lo tengas instalado, puedes comenzar a usarlo. Puedes jugar con él en su consola interactiva, escribiendo y viendo cómo se ejecuta el código en tiempo real. Si prefieres escribir programas más completos, solo necesitas escribir el código en un archivo de texto y ejecutarlo. Además, si estás buscando una forma más avanzada de trabajar, hay programas como PyCharm, Visual Studio Code o Spyder que te ayudaran a

escribir, corregir y ejecutar tu código de manera más eficiente. ¡Así que tienes muchas opciones para comenzar con Python!

Python se ha convertido en un lenguaje de programación realmente importante en el mundo de la ciencia de datos y el aprendizaje automático. La razón detrás de su éxito es que es fácil de aprender y viene equipado con muchas herramientas y bibliotecas que son geniales para trabajar con datos y crear modelos de aprendizaje automático. Piensa en bibliotecas como NumPy, Pandas y Matplotlib como tus herramientas para organizar y mostrar datos, mientras que, para crear y entrenar modelos, podrías utilizar Scikit-learn, TensorFlow y Keras. Pero eso no es todo, Python también está detrás del análisis de datos en tiempo real, la exploración de datos y la creación de inteligencia artificial.

Mathplotlib

Dedicada a realizar gráficas de una manera sencilla, de calidad y como un recurso abierto al público que suele usar el lenguaje de programación Python indicado en Barrett *et al.* (2005).

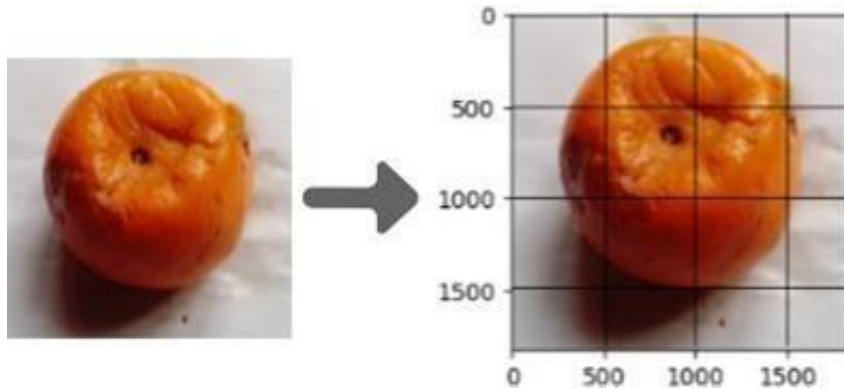
Su arquitectura conceptual de código esta dado por dar funciones que permitan crear gráficas en una línea de comando.

Para recalcar la imagen de las fotos y controlarlas como datos, se puede usar el módulo Pyplot que es el que crea un trazado a los datos orientado a objetos.

```
import matplotlib.pyplot as plt
```


Figura 13

Proceso de imagen luego de usar Matplotlib



Fuente: Autor

La estructura en Matplotlib figure ayuda con los componentes que los diferentes gráficos traen; un poco del glosario que se tiene se habla de figure visto como el área de trazado como cuadrícula en una imagen siendo cada cuadro una subparcela ósea región; para ello puede verse la figura 13 (Yu *et al.*, 2017).

Numpy

Es un paquete de python para computación numérica especializada en procesamiento numérico a través de arreglos multidimensionales; los fundamentos de algebra lineal cuando se habla de matrices se hace de manera eficaz y dinámica sin la preocupación de si por ejemplo quiere cambiar un elemento de la matriz tenga que quitarla complemente al contrario puede aplicar una máscara (Bressert, 2012).

```
import numpy as np
```

Python almacena los datos de diferentes formas, pero entre las más conocidas las listas y los diccionarios, la operación de elementos en las listas se da a través de bucles iterativos siendo muy difícil para python realizarlo solo, a lo que allí es cuando mediante los N-arreglos (Objetos de almacenamiento de datos) actúa Numpy con su velocidad y eficacia de efecto (Harris *et al.*, 2020).

Las imágenes llegan a ser datos que se almacenan mediante listas, siendo los datos matrices que al querer interpretarlos se usa Numpy, entendiendo así el tipo de datos, forma y avances.

Dataset

Es el conjunto de datos como se puede ver en la figura 14 que bien son recolectados a partir de una idea planteada para que luego estos datos sean transformados por una serie de pasos dentro del trabajo y así se arroje un resultado adecuado, para ello debe ser preparado el Dataset con la selección correspondiente a través de técnicas y entrenamiento de datos.

Torres (2018) informa que se debe incluir un conjunto de datos de prueba para realizar luego una validación del modelo que se propuso.

Figura 14

Ejemplo del Dataset – a. Residuos orgánicos reutilizables, b. Residuos orgánicos Compostaje



Fuente: Autor

Keras

Keras viene dado como un recurso para desarrollar el modelo de aprendizaje profundo, que puede trabajar sobre Tensorflow desarrollando un código en menos de 15 líneas funcional donde se puede precargar los datos ya ubicados y puestos a disposición para el modelo a diseñar (Moolayil *et al.*, 2019).

Keras ofrece modelos de aprendizaje profundo con pesos pre entrenados para predicción, extracción de características y ajuste fino. Los pesos se descargan automáticamente y se almacenan en una ubicación predeterminada. Los modelos se construyen según el formato de datos establecido en la configuración de Keras, como. Alto-Ancho-Profundidad “para TensorFlow.

TensorFlow

TensorFlow es una biblioteca de software de aprendizaje automático (machine learning) y redes neuronales desarrollada por Google que hace uso de los tensores matemáticos para realizar sus operaciones.

En TensorFlow, los tensores se utilizan para representar los datos de entrada, los parámetros del modelo y los resultados de las operaciones. TensorFlow también utiliza los tensores para realizar las operaciones de aprendizaje automático, como la multiplicación de matrices y el cálculo de gradientes, que son la base de muchos algoritmos de aprendizaje profundo.

TensorFlow proporciona una amplia variedad de operaciones tensoriales, que permiten a los desarrolladores realizar operaciones matemáticas y de manipulación de datos de manera eficiente en grandes conjuntos de datos. Los tensores en TensorFlow son objetos multidimensionales que pueden tener cualquier número de dimensiones y tipos de datos, como números enteros, flotantes y booleanos.

Además, TensorFlow permite a los usuarios construir redes neuronales de manera fácil y flexible, utilizando la arquitectura de grafos computacionales. En esta arquitectura, las operaciones matemáticas se representan como nodos del grafo, mientras que los datos se representan como aristas. Esto permite a TensorFlow optimizar la ejecución del modelo y realizar cálculos en paralelo en múltiples procesadores y dispositivos, como CPUs, GPUs y TPU (Tensor Processing Units).

En resumen, TensorFlow hace un uso extensivo de los tensores matemáticos para realizar operaciones de aprendizaje automático y construir redes neuronales de manera flexible y eficiente. Esto ha permitido a los desarrolladores y científicos de datos construir modelos de

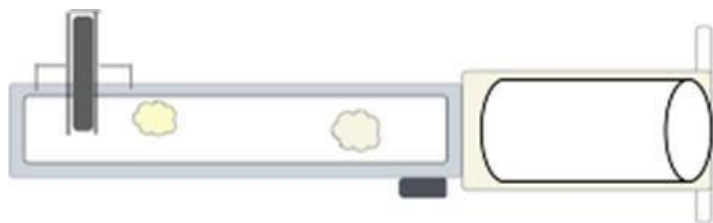
aprendizaje profundo más complejos y precisos para resolver problemas en una amplia variedad de áreas, como la visión por computadora, el procesamiento del lenguaje natural y la robótica.

Algoritmo Basado en Deep Learning para la Clasificación de Residuos Orgánicos

Mediante el artículo dado por Payoma *et al.* (2022) como el diseño y validación de una clasificación mecánica para residuos orgánicos municipales, o el objetivo de mitigar el impacto ambiental en el municipio de Sopo, Cundinamarca - Colombia. Se llega a confirmar el haber escogido los residuos orgánicos en este proyecto y esto por el aumento que ha causado estos de emisión de gases contaminantes, especialmente en metano, lo que lleva a la necesidad de generar energías limpias y renovables. El trabajo utiliza técnicas de reconocimiento de imágenes y redes neuronales para clasificar los residuos orgánicos e inorgánicos por lo que podría complementarse con el algoritmo que se está dando aquí ya que como se menciona en sus resultados se podría actualizar el sistema figura 15 con el uso de redes neuronales convolucionales que permitiría una mayor precisión.

Figura 15

Diseño del sistema mecánico de clasificación de residuos orgánicos.



Nota: Está basado en el diseño del sistema de Payoma *et al.* (2022)

Se utiliza TensorFlow para entrenar un modelo de clasificación de imágenes. Primero, se muestra algunas imágenes del conjunto de datos. Luego, utiliza ImageDataGenerator para aplicar aumentos de datos a las imágenes. A continuación, crea generadores de datos para los conjuntos de entrenamiento y pruebas. Calcula los pesos de clase balanceados y construye un modelo utilizando la arquitectura VGG19. Compila el modelo con una función de pérdida y métricas adecuadas, y finalmente lo entrena utilizando los generadores de datos. Grafica la precisión y pérdida durante el entrenamiento para evaluar el rendimiento del modelo.

El primer paso es asegurarse de que la biblioteca Tensorflow está disponible para su uso. Una de las características destacadas de TensorFlow es su capacidad para aprovechar las GPUs⁵ y acelerar los cálculos necesarios para el entrenamiento y la ejecución de modelos de aprendizaje automático.

Para verificar el número de GPUs disponibles en el sistema se ha utilizado la siguiente función:

```
tf.config.list_physical_devices( 'GPU' )
```

Con esta función, se obtendrá una lista de los dispositivos físicos disponible en el sistema como las GPUs. Al contar los elementos de esta lista utilizando “*len()*” de python, podrás saber cuántas GPUs están disponibles en el sistema. El tener más de una GPU disponible en el sistema ayuda a realizar la tarea de aprendizaje automático en cuanto al entreno y ejecución de los modelos.

⁵ Las Unidades de Procesamiento de Gráficos, comúnmente conocidas como GPUs (Graphics Processing Units), son componentes especializados en el procesamiento y renderizado de gráficos y datos visuales en dispositivos electrónicos (Cebri'n *et al.*, 2012).

Se importa también las bibliotecas como OS⁶ y matplotlib.pyplot⁷ para poder luego comenzar con la preparación de las imágenes. Antes de mostrar las imágenes, se debe asegurar el tener acceso a las imágenes, ya sean que estén almacenadas en carpetas en el escritorio o subidas a drive. Utilizando la función “*os.listdir*” se obtiene una lista de los nombres de los archivos de imagen en esa carpeta. Ahora se muestran las imágenes usando pyplot donde se usa un bucle⁸ para recorrer los nombres de los archivos de imagen y utilizar la función “*mpimg.imread*” para leer cada imagen y almacenarla en una variable. De esa forma se pueden mostrar como ejemplo las primeras 25 imágenes de la carpeta, lo que para organizarlas en cuadrícula de 5×5 se utiliza la función “*plt.subplot*” (Hunter, 2007).

La clase ImageDataGenerator de la biblioteca Tensorflow es importante para el concepto de aumento de datos que se utiliza aquí ya que se recuerda que el aumento de datos es útil para aumentar la variabilidad del conjunto de datos y evitar el sobreajuste de estos, incluso mejorar el rendimiento de los modelos, de ese modo lo primero que se debe hacer es importar también algunas bibliotecas necesarias para ellos, las cuales son:

```
from tensorflow.keras.preprocessing import ImageDataGenerator
```

```
import numpy as np
```

⁶ Son programas fundamentales que actúan como intermediarios entre el hardware de una computadora y las aplicaciones o software que se ejecutan en ella. En esencia, son el “núcleo” del funcionamiento de cualquier dispositivo electrónico, desde computadoras personales hasta teléfonos inteligentes y servidores (López Get, 2010).

⁷ Es un módulo de la biblioteca Matplotlib, que es ampliamente utilizada en el lenguaje de programación Python para crear visualizaciones gráficas de datos. matplotlib.pyplot proporciona una interfaz de programación similar a la de MATLAB, lo que hace que la creación de gráficos sea más sencilla y accesible para los usuarios que están familiarizados con MATLAB (Hunt, 2019)

⁸ Se refieren a estructuras de programación que permiten repetir un conjunto de instrucciones varias veces. Estos bucles son fundamentales para procesar y analizar grandes cantidades de datos visuales de manera eficiente (Kruger *et al.*, 2012).

La biblioteca ImageDataGenerator proporciona las funciones necesarias para generar muestras de datos aumentados, y numpy se utiliza para realizar cálculos y manipulaciones numéricas. Se crea un generador de datos utilizando la clase ImageDataGenerator lo cual aplicara transformaciones a las imágenes de entrada y genera nuevas muestras de datos, lo que podrían ser las ya vistas antes como rotación, desplazamiento, voltear, entre otras transformaciones (Harris *et al.*, 2020).

Una vez ya se tiene el generador de datos, se puede generar conjuntos de datos aumentados. Esto se logra con la función “*flow_from_directory*” del generador de datos donde esta función toma como argumento el directorio que contiene las imágenes de origen y realiza la generación de las nuevas muestras de datos aumentados. Para el código se utilizó el directorio “*./dataset*” para generar conjuntos de datos de entrenamiento y prueba. Se especifica el tamaño objetivo de las imágenes, el tamaño de lote, la opción de mezcla aleatoria de las muestras y se divide el conjunto de datos en entrenamiento y prueba utilizando el parámetro “*subset*”.

Por medio de la biblioteca “*scikit – learn*” se puede calcular los pesos de clase balanceados, siendo esto un problema a abordar cuando las clases del conjunto de datos están desequilibradas, lo que puede afectar el rendimiento del modelo de aprendizaje automático. Las clases únicas y la cantidad de muestras se obtienen de cada clase en el conjunto de datos de entrenamiento. Luego, se utiliza la función “*compute_class_weight*” para calcular los pesos de clase balanceados basados en las frecuencias de clase. Por último, se imprimen los pesos de clase balanceados obtenidos. Estos pesos pueden ser utilizados durante el entrenamiento del modelo para dar más importancia a las clases minoritarias y compensar el desequilibrio en el conjunto de datos lo que al imprimir estos pesos, se proporciona información útil sobre como ajustar el

modelo para tratar adecuadamente las clases desequilibradas y mejorar el rendimiento general del modelo.

```
print ("Pesos de clase balanceados:" , classweights )
```

La forma más común de la asignación de pesos de clase es utilizando un diccionario en este caso se crea uno siendo este:

```
class weights = {i: class weights [ i ] for i in range ( len ( c l a s e s ) ) }
```

Este código crea un diccionario donde las clases son los índices de las clases (0, 1, 2, ..) y los valores son los pesos correspondientes obtenidos anteriormente.

Ahora por medio de la importación de TensorFlow como biblioteca en Python, se puede cargar el modelo VGG19 utilizando la función “*tf.keras.application.VGG19*” se especifica el tamaño de entrada de las imágenes utilizando el argumento “*inputshape = (224,224,3)*”.

Para que el modelo descargado en cuanto a sus pesos y parámetros no se actualicen durante el entrenamiento y se aproveche el conocimiento previo del modelo, lo mismo que evitar un sobreajuste y mejoremos la eficiencia computacional se establece “*net.trainable = false*” (Chollet, 2015).

Se presenta un modelo secuencial en TensorFlow utilizando la clase “*tf.keras.Sequential*”.

Este modelo consta de varias capas apiladas en secuencia como lo son:

- El “*net*” representa el modelo que está cargado, a lo que al finalizar el modelo secuencial se aprovechan las características aprendidas para la clasificación de imágenes.
- Para regularizar el modelo se usa “*tf.keras.layers.Dropout(0,5)*”, recordando que esto también se hace con el fin de prevenir el sobreajuste, por lo que la técnica de dropout queda como la ayuda necesaria realizando la desactivación aleatoria de una fracción de las unidades de salida durante el entrenamiento.
- La capa densa de salida es “*tf.keras.layers.Dense(2, activation = 'sigmoid')*”; al tener 2 unidades lo que se informa es el modelo que tiene una clasificación binaria. Aquí se ve la función de activación dada como “*Sigmoid*”, utilizándose para obtener el rango de salida de 0 a 1, siendo esta la representación como la probabilidad de pertenecer a una o a la otra clase.

Continuando con las funciones tenemos que luego tener en cuenta la función “*modelo.summary()*”, siendo esta el resumen de lo que ha estado pasando con el modelo definido. Esta proporciona información como el tipo de capa, la forma de entrada y de salida, el número de parámetros entrenables y el total de estos en el modelo.

La función “*compile()*” indica la compilación del modelo en TensorFlow, esto con la idea de configurar el modelo con el optimizador “*optimizer = 'adam'*” siendo un algoritmo usado para ajustar los pesos y parámetros del modelo durante el entrenamiento; la función de pérdida “*loss = 'categorical_crossentropy'*”, como métrica que mide la discrepancia entre las predicciones del modelo y las etiquetas verdaderas durante el entrenamiento y por último las

métricas que llegan a ser mediadas adicionales utilizadas para evaluar el rendimiento del modelo durante su entrenamiento, pero no influyen directamente en la optimización del modelo.

El proceso de entrenamiento comienza con el modelo ajustando sus parámetros en este caso se debe establecer el número de épocas en la variable “EPOCAS” representando la cantidad de veces que la modelo vera el conjunto de datos. Luego utilizando el conjunto de datos de entrenamiento “*datagen – entrenamiento. a través de las ‘epocas ‘epochs = EPOCAS’*” definidas ya anteriormente, se tiene en cuenta la representación del tamaño de lote que es la cantidad de ejemplos que el modelo procesara en cada paso y la evaluación “*validationdata = datagenpruebas*” que proporciona un diccionario que contiene los pesos de clase balanceados (Torres, 2018).

Para traficar los resultados y como tal evaluar el rendimiento, así mirar si el modelo está entrenando correctamente, se realiza un código que ha de realizar dos graficas donde en una se encuentra la precisión de entrenamiento y prueba en función del número de épocas:

```
plt.plot ( rango epocas, acc, label = ' Precision Entrenamiento ' )
plt.plot ( rango epocas, val acc, label = ' Precision Pruebas ' )
```

Y la otra muestra la perdida de entrenamiento y prueba en función del número de épocas:

```
plt . plot ( rango epocas , loss, label = ' Perdida de entrenamiento ' )
plt . plot ( rango epocas , val loss, label = ' Perdida de pruebas ' )
```

Estas gráficas son la herramienta útil para visualizar el progreso y rendimiento del modelo durante el proceso de entrenamiento.

Análisis de Resultados

En este capítulo, se presenta los resultados obtenidos al entrenar y validar diferentes arquitecturas de redes neuronales profundas para la clasificación de residuos orgánicos. Se analiza como estas redes se comportan en la tarea de identificar y categorizar residuos orgánicos entre reutilizable y compostaje, a parte de los datos de entrenamiento y validación utilizados en el estudio.

Por el trabajo de investigación, se ha llevado a cabo la implementación y evaluación de nuestro enfoque utilizando la metodología detallada en el Problema de Investigación. A medida que se recolectaban las imágenes de residuos orgánicos, se creó un conjunto de datos inicial. Luego, se aplicaron técnicas de procesamiento y etiquetado a estas imágenes. A continuación, se procedió a experimentar con diversas arquitecturas de redes neuronales profundas explicadas en el capítulo denominado Fundamentación Teórica. Durante esta fase, se evaluó el rendimiento de cada arquitectura en la tarea de clasificación de residuos orgánicos. Estas experimentaciones nos permiten aumento de datos como técnica para permitir el aumento de la cantidad de imágenes de los datos que ya se tenían como existentes, aplicándose rotaciones en un rango de 30 grados, desplazamientos horizontales y verticales aleatorias a las imágenes, se aplica deformación de cizallamiento aleatoria a las imágenes con un rango de ± 20 , al igual que la aplicación de un zoom y división con el mismo valor, por último, un volteo horizontal aleatorio.

Durante las etapas de entrenamiento y prueba, en lugar de utilizar nuestra propia computadora, optamos por utilizar Google Colab, específicamente en su versión gratuita. La elección de esta plataforma se basó en su gran capacidad de cálculo computacional, gracias a la disponibilidad de unidades de procesamiento gráfico (GPUs) (Carneiro *et al.*, 2018).

Según BODERO et al. (2020), Google Colab es una alternativa para el procesamiento de una red neuronal convolucional para la clasificación de imágenes, demostrando que no existe una diferencia significativa en el tiempo de entrenamiento de la red neuronal en Google Colab y un computador personal, sin embargo, existe una menor pérdida y mayor precisión del modelo en la clasificación de imágenes.

Las métricas usadas, se tuvo en cuenta lo que se dice en la sección llamada Métricas para clasificadores de aprendizaje automático, lo que al realizar una explicación a detalle porque se ha escogido decimos que la Precisión (Accuracy) indica la proporción de predicciones correctas realizadas por el modelo con respecto al total de predicciones. Esta no solo se presenta en entrenamiento sino también la validación del

modelo, lo que hace una doble funcionalidad. La pérdida (Loss) es la métrica que representa la discrepancia entre las predicciones del modelo y las etiquetas verdaderas durante el entrenamiento ósea como se ajusta los datos de entrenamiento a lo largo de las épocas.

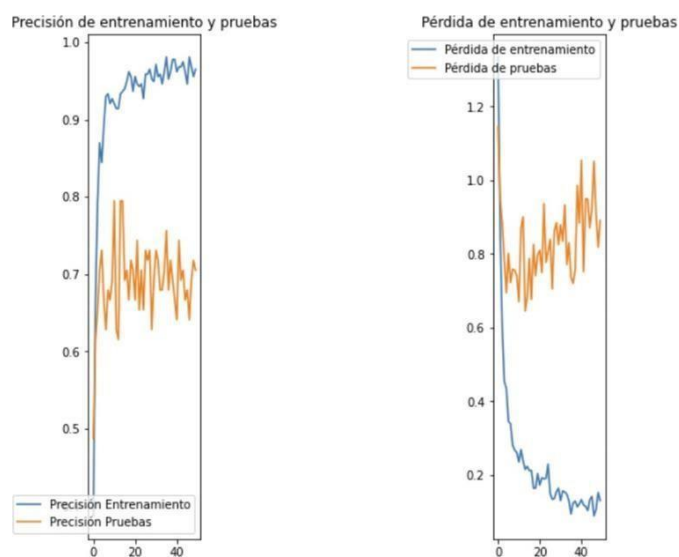
En el capítulo titulado Algoritmo basado en Deep Learning para la clasificación de residuos orgánicos donde se enmarca la configuración del modelo con el optimizador, para lo que se escogió el algoritmo de optimización más popular y con algunos proyectos similares en el campo de aprendizaje profundo (Géron, 2022). Este optimizador lo que utiliza es una tasa de aprendizaje adaptable que varía según el historial de las gradientes de los parámetros del modelo, permitiéndose así que el algoritmo ajuste los pesos de manera más efectiva para converger hacia el mínimo local de la función de pérdida.

A continuación, se detalla las pruebas realizadas:

Una vez validado el modelo se corrió ya capturados los datos y procesados validando el entrenamiento de la red. La salida obtenida en línea de código TensorFlow Board, se representa a continuación con dos gráficas vistas en la figura 16. Para leer la figura 16, se indica que la gráfica a la izquierda se muestra las precisiones obtenidas en el modelo para los datos en cuanto a entrenamiento en color naranja y pruebas en color azul. La gráfica a la derecha muestra las pérdidas obtenidas de entrenamiento y en pruebas. Ambas en la parte inferior continúan la numeración de las épocas en la muestra.

Figura 16

Primeras pruebas del algoritmo y sus resultados en cuanto a su precisión y pérdida del modelo.

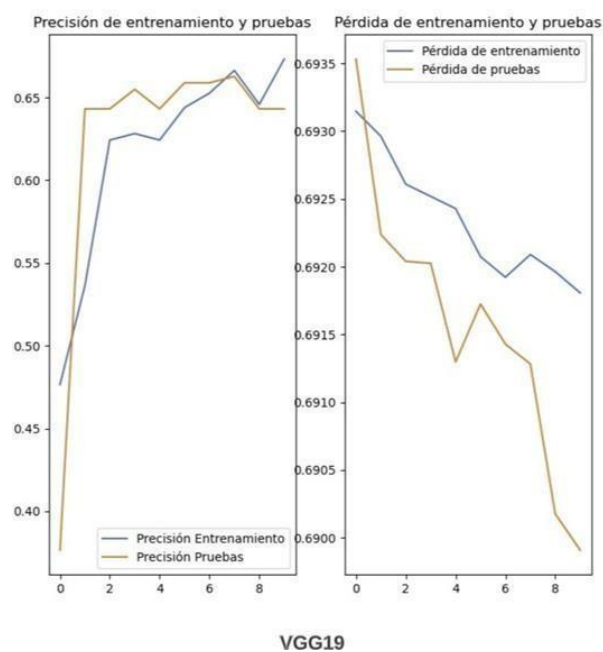


Fuente: Autor

De primera mano, el programa realizaba el objetivo de clasificación, pero no obtenía las gráficas esperadas, lo que se tomó como un sobreajuste del modelo, el cual se intentó en reiteradas ocasiones algunos cambios en su configuración, pero al final se modificó agregando la

técnica de dropout y pesos de clases balanceados, mostrando mejor el comportamiento del modelo en la gráfica como se muestra en la figura 17.

Figura 17
Modelo corrido con arquitectura VGG19 a 10 épocas

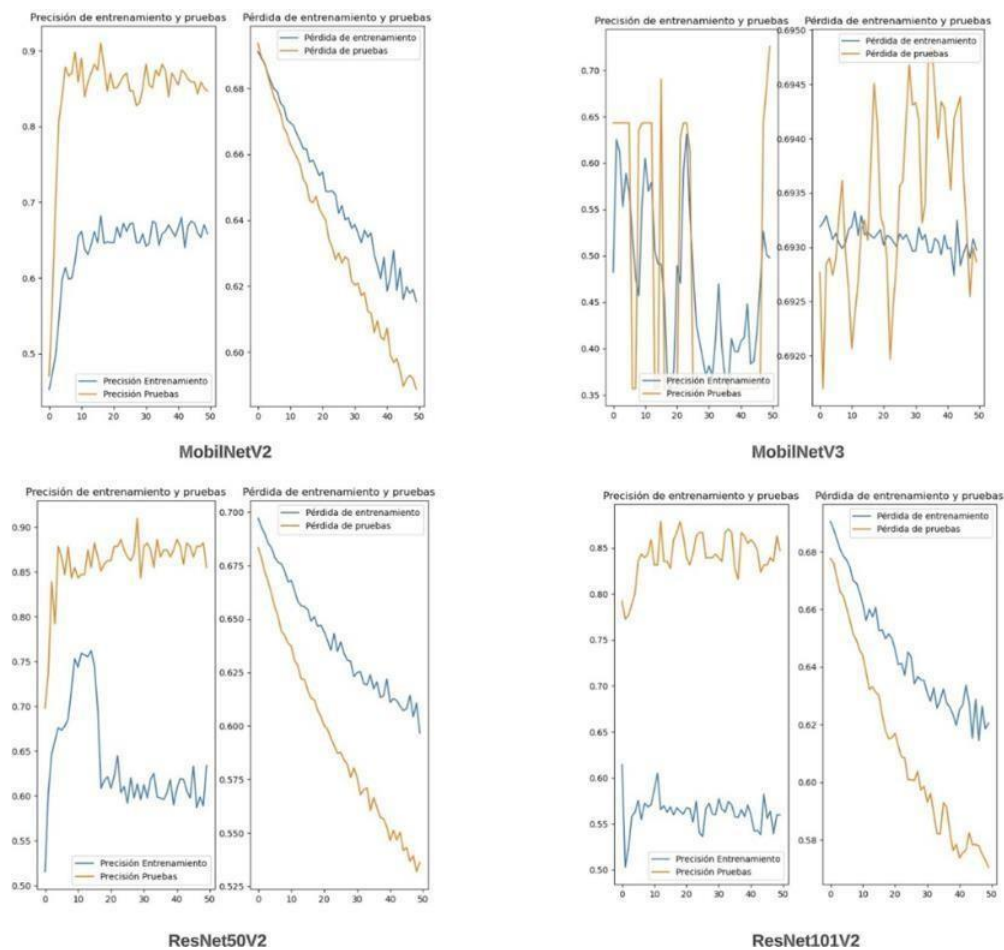


Fuente: Autor

Se corrieron las seis arquitecturas con el modelo, usando 50 épocas cada una, lo que se pueden ver algunos de los resultados en la figura 18, para lo que se toma como análisis de que entre más cercanas estén, quiere decir que el rendimiento es comparable tanto en el entrenamiento como en los conjuntos de datos de validación. Para lo que en este caso MobileNetV2 y VGG19 presentaron mejores resultados, decidiéndose correr ambas con un número mayor de épocas.

Figura 18

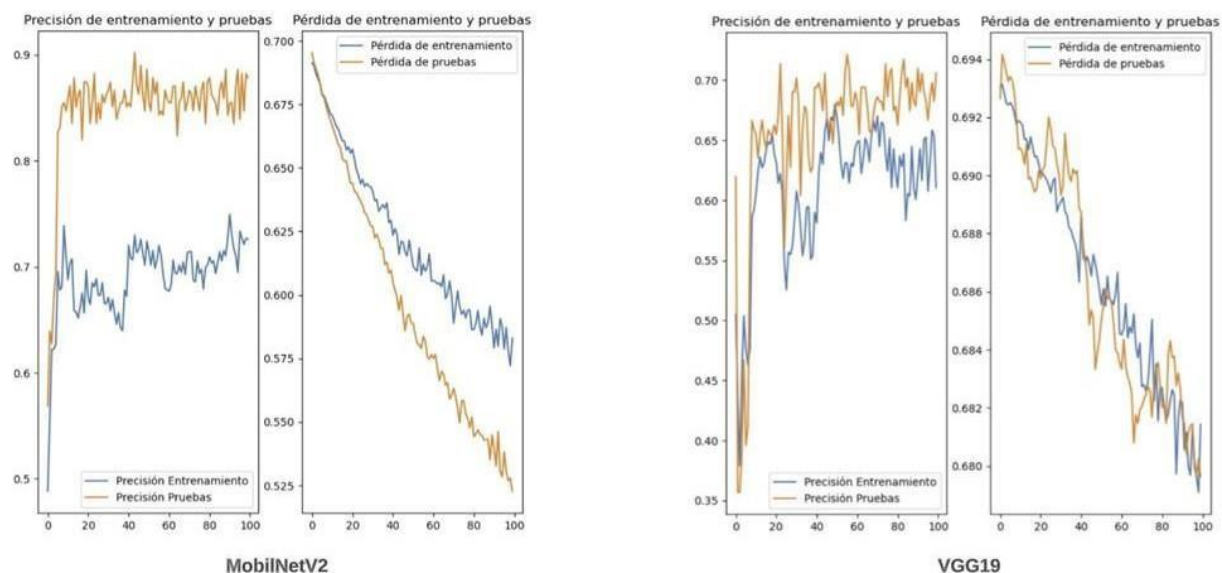
Algunas de las arquitecturas CNN usando el modelo y corriendo a 50 épocas



Fuente: Autor

Al subirse el valor de los épocas a 100 en dos de las arquitecturas y se pudiera presenciar la precisión y mejores hallazgos, MobileNetV2 mostraba un underfitting figura 19, mencionando a Bashir *et al*, (2020) que muestra el subajuste como algo que ocurre cuando un algoritmo de aprendizaje automático carece de suficiente capacidad de modelo o entrenamiento para aprender completamente la verdadera relación entre las características y las respuestas de un conjunto de datos, ya sea a través de la memorización o no.

Figura 19
Arquitecturas MobilNetV2 y VGG19 específicamente, corridas con 100 épocas



Fuente: Autor

Finalmente, la arquitectura de red neuronal convolucional VGG19 al analizar los gráficos generados durante el proceso de entrenamiento y validación muestran una tendencia de convergencia hacia valores estables y bajos respectivamente, lo que indica un aprendizaje adecuado sin signos de sobreajuste ni subajuste. Además, la curva de pérdida de VGG19 muestra una disminución significativa y constante, lo que sugiere que el modelo está ajustando los pesos de manera efectiva para minimizar la discrepancia entre las predicciones y las etiquetas verdaderas.

La arquitectura de CNN VGG19 tiene un valor de precisión de 0.7137 y de pérdida 0.6781, lo que podemos indicar que está haciendo predicciones correctas en más del 70% de los casos durante la evaluación. Aunque la precisión es decente, la pérdida de 0.6781 indica que el modelo todavía tiene margen de mejora en términos de la diferencia entre las predicciones y los valores reales durante el proceso de entrenamiento ya que, si se tiene en cuenta en la gráfica de VGG19 en la figura 19 de pérdida, se puede ver un descenso lineal lo que todavía no se observa

la velocidad de convergencia sobre épocas o estabilidad para el buen ajuste del modelo y para que se muestre la convergencia habría que subir el valor de la época en más de 100.

Conclusiones

Para la presente investigación, existen base de datos cercanos a la propuesta como por ejemplo la clasificación de frutas entre buenas y dañadas; pero no llevan al mismo objetivo; por este motivo, es que se han tenido que recolectar imágenes de residuos orgánicos para poder tener un base de datos con la cual se pueda realizar pruebas.

A medida que se avanzaba en el proceso de desarrollo, se hizo algunos cambios en las decisiones iniciales de diseño del algoritmo. En un principio, se pensaba en utilizar únicamente TensorFlow como la biblioteca principal para redes neuronales. Sin embargo, en versiones posteriores, decidimos también incluir Keras por dos razones: primero, porque ya estaba integrada en la biblioteca TensorFlow; y segundo, porque proporciona un enfoque más amigable y abstracto para programar en comparación con TensorFlow.

Se evidencia que no solo se puede tener un buen resultado del modelo por medio de escoger la arquitecturas de red neuronal convolucional indicada sino también de una buena recolección de datos, en este caso el conjunto de imágenes dado como dataset, no se tenía la cantidad de imágenes necesarias para correr, por eso fue al no tenerse una base muy grande, se hizo el aumento de imágenes, esta como ayuda para cuando no, lo que luego sería crítico realizar un balance con el peso de clases para el modelo.

Como se mencionó en capítulo correspondiente a Análisis de resultados, el uso de algunas arquitecturas que tenían como objetivo predecir el estado de los residuos orgánicos entre reutilizables y de compostaje, a lo que es importante dejar en evidencia que algo que se puede recalcar es la idea de que entre más épocas tenga el modelo al correrlo, mejores resultados se presentan. De las seis arquitecturas, el modelo que tuvo mayor aceptación con la arquitectura de

red neuronal convolucional es VGG19. Esto sugiere que la red neuronal convolucional VGG19 ha logrado un rendimiento aceptable en la tarea de clasificación de residuos orgánicos ya que esta no presenta un sobreajuste, sin embargo puede seguir mejorando si aumentamos las épocas para ver cómo se va tranzando con los datos de entrenamiento, para ello se tendría que tener posiblemente aumentar el conjunto de datos y tener una mayor capacidad de memoria a lo que en este momento con el acceso de Google Colab en su versión gratis no se podría.

En relación con las consideraciones que afectan a proyectos con enfoque similar, es importante mencionar que Google Colab puede imponer ciertas restricciones a los investigadores debido a limitaciones en los tiempos y recursos computacionales. Por ejemplo, cuando se busca compartir y permitir que otros realicen modificaciones en tiempo real, se pueden encontrar obstáculos debido a las limitaciones de interacción y control en el entorno colaborativo de Colab. Esto puede dificultar la colaboración en tiempo real y la capacidad de realizar ajustes rápidos en los modelos.

Al examinar minuciosamente la documentación pertinente sobre la clasificación de residuos orgánicos y las redes neuronales artificiales, se revela que los estudios previos datan de años anteriores. Este aspecto cobra relevancia al considerar el adiestramiento de redes neuronales convolucionales, como se ejemplifica en la investigación de Uddin and Basal (2021), que varios de los artículos tomados como referentes se basan en arquitecturas como VGG16. No obstante, en ese período, la arquitectura VGG19 aún no había sido concebida ni actualizada, lo que subraya la necesidad de incorporar en futuros modelos las iteraciones más recientes que emergen y se documentan en las subsiguientes entregas de dicho compendio.

La investigación aquí adscrita ha dado exclusividad al objetivo dando más del 70%, con este resultado se puede dar aun así una buena precisión del modelo.

Referencias Bibliográficas

- Abdel-Shafy, H. I., & Mansour, M. S. (2018). Solid waste issue: Sources, composition, disposal, recycling, and valorization. *Egyptian journal of petroleum*, 27, 1275–1290.
- Acevedo, E., Serna, A., & Serna, E. (2017). Principios y características de las redes neuronales artificiales. *Desarrollo e innovación en ingeniería*, 173.
- ACOSTA-GUZMAN, I. L., VARELA-TAPIA, E. A., ACOSTA-VARELA, C. I., & TUMBACO-BRAVO, J. D. (n.d.). Predicción de índice de masa corporal (IMC) mediante algoritmos de Máquina de Soporte Vectorial y Árboles de Decisión de Machine Learning de la IA.
- Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. *2017 international conference on engineering and technology (ICET)*, (pp. 1–6).
- Al-Khatib, I. A., Monou, M., Zahra, A. S., Shaheen, H. Q., & Kassinos, D. (2010). Solid waste characterization, quantification and management practices in developing countries. A case study: Nablus district–Palestine. *Journal of environmental management*, 91, 1131–1138.
- Altikat, A. A., Gulbe, A., & Altikat, S. (2022). Intelligent solid waste classification using deep convolutional neural networks. *International Journal of Environmental Science and Technology*, 1–8.
- Alvarado, R. A., Payoma, F. A., Andrés, C., Cárdenas, V., Riaños, C. K., Galeano, Z. V., & Quintero, L. D. (n.d.). Revisión sistemática de publicaciones con énfasis en Ciencia Tecnología e Innovación del municipio de Sopó periodo.

- Antona Castañares, A. (2020). *Aplicación del dropout a la cuantificación de la incertidumbre en redes neuronales*. Ph.D. dissertation, Industriales.
- Barrett, P., Hunter, J., Miller, J. T., Hsu, J.-C., & Greenfield, P. (2005). matplotlib—A Portable Python Plotting Package. *Astronomical data analysis software and systems XIV*, 347, p. 91.
- Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An information-theoretic perspective on overfitting and underfitting. *AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, AI 2020, Canberra, ACT, Australia, November 29–30, 2020, Proceedings 33*, (pp. 347–358).
- Biswas, A., Nasim, M. D., Imran, A., Sejuty, A. T., Fairouz, F., Puppala, S., & Talukder, S. (2023). Generative Adversarial Networks for Data Augmentation. *arXiv preprint arXiv:2306.02019*.
- Boden, M. A. (2017). *Inteligencia artificial*. Turner.
- BODERO, E. M., LOPEZ, M. P., CONGACHA, A. E., CAJAMARCA, E. E., & MORALES, C. H. (2020). Google Colaboratory como alternativa para el procesamiento de una red neuronal convolucional. *Revista Espacios*, 41.
- Bonaccorso, G. (2017). *Machine learning algorithms*. Packt Publishing Ltd.
- Bressert, E. (2012). SciPy and NumPy: an overview for developers.
- Carneiro, T., Da Nóbrega, R. V., Nepomuceno, T., Bian, G.-B., De Albuquerque, V. H., & Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6, 61677–61685.

- Cebri'n, J. M., Guerrero, G. D., & Garcia, J. M. (2012). Energy efficiency analysis of GPUs. *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, (pp. 1014–1022).
- Cerdá, E., & Khalilova, A. (2016). Economía circular. *Economía industrial*, 401, 11–20.
- Chanampe, H., Aciar, S., Vega, M. d., Molinari Sotomayor, J. L., Carrascosa, G., & Lorefice, A. (2019). Modelo de redes neuronales convolucionales profundas para la clasificación de lesiones en ecografías mamarias. *XXI Workshop de Investigadores en Ciencias de la Computación (WICC 2019, Universidad Nacional de San Juan)*.
- Chollet, F. (2015). keras. *keras*. GitHub.
- Duran, G. C., Sato, A. K., Ueda, E. K., Takimoto, R. Y., Martins, T. C., & Tsuzuki, M. S. (2021). Electrical impedance tomography image reconstruction using convolutional neural network with periodic padding. *IFAC-PapersOnLine*, 54, 418–423.
- El Naqa, I., & Murphy, M. J. (2015). *What is machine learning?* Springer.
- Erickson, B. J., & Kitamura, F. (2021). Magician's corner: 9. Performance metrics for machine learning models. *Magician's corner: 9. Performance metrics for machine learning models*, 3, e200126. Radiological Society of North America.
- Feng, J., & Lu, S. (2019). Performance analysis of various activation functions in artificial neural networks. *Journal of physics: conference series*, 1237, p. 022030.
- Galdames, O. N. (2016). *Efectividad del té de compost en el control de Botrytis cinerea Pers. ex Fr. en arándanos (Vaccinium corymbosum L.) cv. Duke, durante floración*. Ph.D. dissertation, Universidad Austral de Chile.

- GALIPIENSO, A. L., ISABEL, M. A., Cazorla Quevedo, M. A., Colomina Pardo, O., Escolano Ruiz, F., & LOZANO ORTEGA, M. A. (2003). *Inteligencia artificial: modelos, técnicas y áreas de aplicación*. Ediciones Paraninfo, SA.
- Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. "O'Reilly Media, Inc."
- Gilbert, J. K. (1995). Educación tecnológica: una nueva asignatura en todo el mundo. *Enseñanza de las ciencias: revista de investigación y experiencias didácticas*, 15–24.
- González Duque, R. (2011). Python para todos. *Python para todos*. Creative Commons Reconocimiento.
- Gonzalez, L. V., Gómez, S. P., & Abad, P. A. (2017). Aprovechamiento de residuos agroindustriales en Colombia. *RIAA*, 8, 141–150.
- Guo, H.-n., Wu, S.-b., Tian, Y.-j., Zhang, J., & Liu, H.-t. (2021). Application of machine learning methods for the prediction of organic solid waste treatment and recycling processes: A review. *Bioresource technology*, 319, 124114.
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., . . . others. (2020). Array programming with NumPy. *Nature*, 585, 357–362.
- Hasan, M. Z., Zahan, N., Zeba, N., Khatun, A., & Haque, M. R. (2021). A Deep Learning-Based Approach for Potato Disease Classification. *Computer Vision and Machine Learning in Agriculture*, 113–126.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, (pp. 630–645).
- Herrera, A. O. (1979). Desarrollo, medio ambiente y generación de tecnologías apropiadas.

- Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., . . . others. (2019). Searching for mobilenetv3. *Proceedings of the IEEE/CVF international conference on computer vision*, (pp. 1314–1324).
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9, 90–95.
- Ismail Jabiullah, M., Ahammed, S., Rahman, M., Hasan, M., & Ahmed, K. (2021). Implementation of a Convolutional Neural Network for the Detection of Tomato-Leaf Diseases. *Computer Vision and Machine Learning in Agriculture*, 87–98.
- Jana, S., Parekh, R., & Sarkar, B. (2021). Detection of Rotten Fruits and Vegetables Using Deep Learning. *Computer Vision and Machine Learning in Agriculture*, 31–49.
- Kabir, M. M., Ohi, A. Q., & Mridha, M. F. (2021). A multi-plant disease diagnosis method using convolutional neural network. *Computer Vision and Machine Learning in Agriculture*, 99–111.
- Karlik, B., & Olgac, A. V. (2011). Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1, 111–122.
- Kaza, S., Yao, L., Bhada-Tata, P., & Van Woerden, F. (2018). *What a waste 2.0: a global snapshot of solid waste management to 2050*. World Bank Publications.
- Kotsiantis, S., Kanellopoulos, D., Pintelas, P., & others. (2006). Handling imbalanced datasets: A review. *GESTS international transactions on computer science and engineering*, 30, 25–36.

- Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., . . . Wiskott, L. (2012). Deep hierarchies in the primate visual cortex: What can we learn for computer vision? *IEEE transactions on pattern analysis and machine intelligence*, 35, 1847–1871.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521, 436–444.
- Lloreda, M. J. (2001). Mecanismos de detección del contraste cromático y de luminancia en el sistema visual humano. *Anales de Psicología/Annals of Psychology*, 17, 219–233.
- López Get, A. (2010). La guerra de los sistemas operativos V1. 0.
- López, R. F., & Fernández, J. M. (2008). *Las redes neuronales artificiales*. Netbiblo.
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9, 381–386.
- Martinez Llamas, J. (2018). Reconocimiento de imágenes mediante redes neuronales convolucionales.
- Moolayil, J., Moolayil, J., & John, S. (2019). *Learn Keras for deep neural networks*. Springer.
- Mora, E. A., Huitrón, V. G., Rangel, H. R., & Sosa, L. E. (2021). Detección de enfermedades foliares con arquitecturas de redes neuronales convolucionales. *RINDERESU*, 5.
- Muñiz, A. G. (2018). Aplicaciones de técnicas de inteligencia artificial basadas en aprendizaje profundo (deep learning) al análisis y mejora de la eficiencia de procesos industriales. *Máster en Ingeniería de Automatización e Informática Industrial. Universidad de Oviedo*.
- Nacelle, A., & Mizraji, E. (2009). Redes neuronales artificiales. *Núcleo de ingeniería biomédica– Universidad de la Republica Uruguay*.
- Ncr, P. C., Clinton, J., Ncr, R. K., Khabaza, T., Reinartz, T., Shearer, C., & Wirth, R. (1999). CRISP-DM 1.0.

- Nnamoko, N., Barrowclough, J., & Procter, J. (2022). Solid waste image classification using deep convolutional neural network. *Infrastructures*, 7, 47.
- Ojeda, C. K. (n.d.). Revisión sistemática PRISMA de la clasificación de residuos sólidos mediante aprendizaje automático. PRISMA systematic review of solid waste classification by machine learning.
- O'Shea, K., & Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*.
- Payoma, F. T., Velásquez, B. V., Alvarado, R. R., Aldana, F. H., & Guarín, J. G. (2022). Diseño y validación de una clasificadora mecánica para residuos orgánicos municipales. *Ingenio Magno*, 13, 113–124.
- Pedreño, J. N., Herrero, J. M., Lucas, I. G., & Beneyto, J. M. (1995). *Residuos orgánicos y agricultura*. Universidad de Alicante.
- PÉREZ, D. D. (2014). Ley 20.412 de sistema de incentivos para la sustentabilidad agroambiental de los suelos agropecuarios: análisis de la conservación y remediación de suelos en Chile. *la ley*, 20, 2.
- Pérez, M. A. (2009). Espacios de Color RGB, HSI y sus Generalizaciones a n-Dimensiones. *INAOE, Tonantzintla*.
- Ponti, M. A., & Da Costa, G. B. (2018). Como funciona o deep learning. *arXiv preprint arXiv:1806.07908*.
- Porras, Á. C., & González, A. R. (2016). Aprovechamiento de residuos orgánicos agrícolas y forestales en Iberoamérica. *Academia y virtualidad*, 9, 90–107.
- Qiu, Z., Zhao, S., Feng, X., & He, Y. (2020). Transfer learning method for plastic pollution evaluation in soil using NIR sensor. *Science of The Total Environment*, 740, 140118.

- Rahat, M., Hasan, M., Hasan, M. M., Islam, M. T., Rahman, M. S., Islam, A. K., & Rahman, M. M. (2021). Deep CNN-based mango insect classification. *Computer Vision and Machine Learning in Agriculture*, 67–85.
- Rojas, T. V., Sanz, W., & Arteaga, F. (2008). Sistema de visión por computadora para la detección de objetos esféricos a través de la transformada de Hough. *Revista Ingeniería UC*, 15, 77–87.
- Rouhiainen, L. (2018). Inteligencia artificial. *Madrid: Alienta Editorial*.
- Sakib, S., Ahmed, N., Kabir, A. J., & Ahmed, H. (2019). An overview of convolutional neural network: Its architecture and applications.
- Sánchez-Guashpa, A., Pico-Valencia, P., Jiménez, P., & Holgado-Terriza, J. A. (2021). Sistema de Clasificación Automático de Peces Endémicos del Ecuador Usando Redes Neuronales Convolucionales. *Revista Iberica de Sistemas e Tecnologias de Informacao*, 444–457.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (pp. 4510–4520).
- Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, 6, 310–316.
- Shyam, R. (2021). Convolutional neural network and its architectures. *Journal of Computer Technology & Applications*, 12, 6–14p.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

- Tao, W., Xie, Z., Zhang, Y., Li, J., Xuan, F., Huang, J., . . . Yin, D. (2021). Corn Residue Covered Area Mapping with a Deep Learning Method Using Chinese GF-1 B/D High Resolution Remote Sensing Images. *Remote Sensing*, *13*, 2903.
- Toğaçar, M., Ergen, B., & Cömert, Z. (2020). Waste classification using AutoEncoder network with integrated feature selection method in convolutional neural network models. *Measurement*, *153*, 107459.
- Torres, J. (2018). *DEEP LEARNING Introducci—n pr:ctica con Keras*. Lulu. com.
- Uddin, M. S., & Bansal, J. C. (2021). *Computer Vision and Machine Learning in Agriculture*. Springer.
- Uddin, M. S., & Bansal, J. C. (2021). Introduction to computer vision and machine learning applications in agriculture. In *Computer Vision and Machine Learning in Agriculture* (pp. 1–8). Springer.
- Urbáez, M. F. (2015). La innovación como cultura organizacional sustentada en procesos humanos. *Revista Venezolana de Gerencia*, *20*, 355–371.
- Vargas, J. W., Buzón, J. A., Vergara, D. A., & Molina, E. (2011). Reducción de los residuos sólidos orgánicos en Colombia por medio del compostaje liquido. *Ingeniare*, 37–44.
- Velardo, C., & Dugelay, J.-L. (2012). What can computer vision tell you about your weight? *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, (pp. 1980–1984).
- Villegas-Cornelio, V. M., & Laines Canepa, J. R. (2017). Vermicompostaje: I avances y estrategias en el tratamiento de residuos sólidos orgánicos. *Revista mexicana de ciencias agrícolas*, *8*, 393–406.

- Vo, A. H., Vo, M. T., Le, T., & others. (2019). A novel framework for trash classification using deep transfer learning. *IEEE Access*, 7, 178631–178639.
- Wei, J., Chu, X., Sun, X.-Y., Xu, K., Deng, H.-X., Chen, J., . . . Lei, M. (2019). Machine learning in materials science. *InfoMat*, 1, 338–358.
- Yang, T.-J., Howard, A., Chen, B., Zhang, X., Go, A., Sandler, M., . . . Adam, H. (2018). Netadapt: Platform-aware neural network adaptation for mobile applications. *Proceedings of the European Conference on Computer Vision (ECCV)*, (pp. 285–300).
- Yu, A., Chung, C., & Yim, A. (2017). *Matplotlib 2. x By Example*. Packt Publishing Ltd.