

# Clasificación de Pokémon con Redes Neuronales Convolucionales

*RONY SANTIAGO BAÑO I RICO*

## Introducción

El presente informe detalla el desarrollo de un sistema de clasificación automática de Pokémon basado en redes neuronales convolucionales (CNN). Este proyecto tiene como objetivo predecir el tipo principal y secundario de un Pokémon dado su imagen, utilizando técnicas avanzadas de aprendizaje profundo. El enfoque principal radica en el uso de un modelo preentrenado (MobileNetV2), combinado con técnicas de preprocesamiento de datos y ajuste de hiperparámetros.

## Estructura de los Notebooks

El proyecto se desarrolló en múltiples notebooks, cada uno enfocado en una etapa específica del flujo de trabajo. A continuación, se detalla la estructura de los notebooks entregados:

1. Exploración de Datos: En este notebook se realizó una inspección inicial del dataset, incluyendo la visualización de distribuciones, detección de desbalances entre clases y verificación de valores faltantes. También se generaron gráficos que ilustran la composición del dataset.
2. Preprocesamiento y Generación de Datos: Este notebook incluyó la preparación de los datos, como la normalización de las imágenes, la creación de generadores de datos para el entrenamiento y validación, y la implementación de técnicas de aumento de datos (Data Augmentation).
3. Entrenamiento del Modelo: Aquí se implementó el modelo basado en MobileNetV2, preentrenado en el dataset ImageNet. Se añadieron capas densas para ajustar el modelo al problema específico, y se utilizaron técnicas como regularización L2 y Dropout. Además, se empleó Early Stopping para evitar el sobreajuste.
4. Evaluación de Resultados: Este notebook incluyó el análisis detallado de los resultados del modelo. Se generaron métricas como precisión, recall, F1-score y una

matriz de confusión, así como visualizaciones de ejemplos de predicciones correctas e incorrectas.

### **Descripción de la Solución**

La solución implementada combina técnicas avanzadas de aprendizaje profundo y preprocesamiento de datos. A continuación, se describen los principales componentes de la solución:

#### **Arquitectura del Modelo:**

El modelo utilizado se basa en MobileNetV2, una arquitectura de red neuronal convolucional ligera y eficiente, preentrenada en el dataset ImageNet. MobileNetV2 fue adaptada para este problema al añadir capas densas personalizadas, ajustadas al número de clases del dataset (18 tipos de Pokémon). Se utilizaron técnicas de regularización como Dropout (para prevenir el sobreajuste) y L2 Regularization.

#### **Preprocesamiento de Imágenes:**

Las imágenes fueron redimensionadas a 128x128 píxeles y normalizadas para que los valores de los píxeles estuvieran en el rango [0, 1]. Además, se aplicaron técnicas de aumento de datos (Data Augmentation), incluyendo rotación, desplazamiento, zoom y espejo horizontal, para mejorar la capacidad de generalización del modelo.

#### **Entrenamiento:**

Se utilizó Early Stopping con 5 épocas para detener el entrenamiento si el rendimiento en el conjunto de validación dejaba de mejorar. El modelo fue entrenado durante 20 épocas con un tamaño de lote de 32, utilizando el optimizador Adam con una tasa de aprendizaje inicial de 0.001.

#### **Evaluación:**

La evaluación del modelo se realizó utilizando métricas como precisión, recall, F1-score y una matriz de confusión. Estas métricas permitieron identificar las clases más difíciles de predecir, así como los errores más comunes.

#### **Iteraciones Realizadas**

El desarrollo del modelo incluyó varias iteraciones, cada una con ajustes y mejoras basadas en los resultados obtenidos:

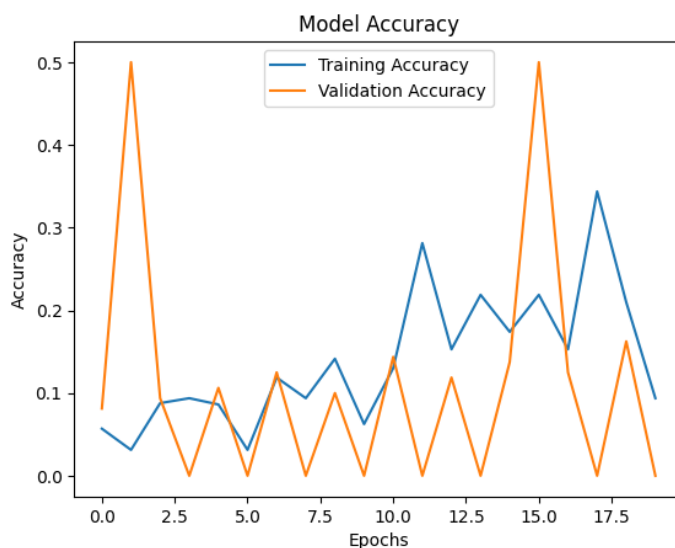
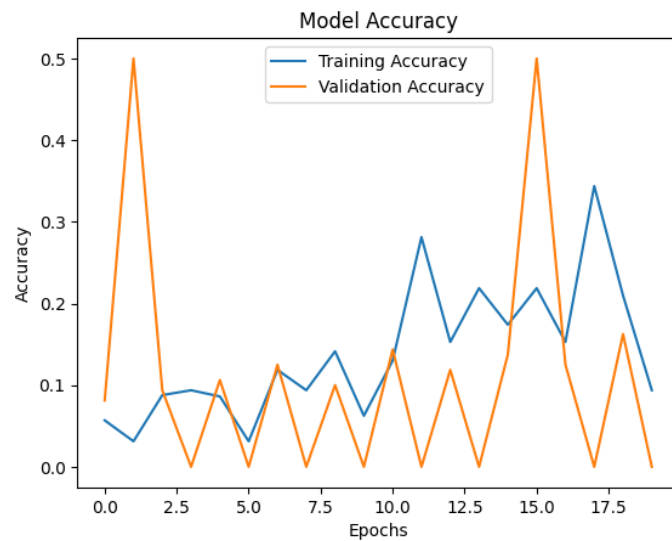
1. Primera Iteración: Se implementó una red neuronal convolucional simple diseñada desde cero. Aunque el modelo mostró cierto potencial, la precisión fue limitada debido a la falta de características avanzadas.

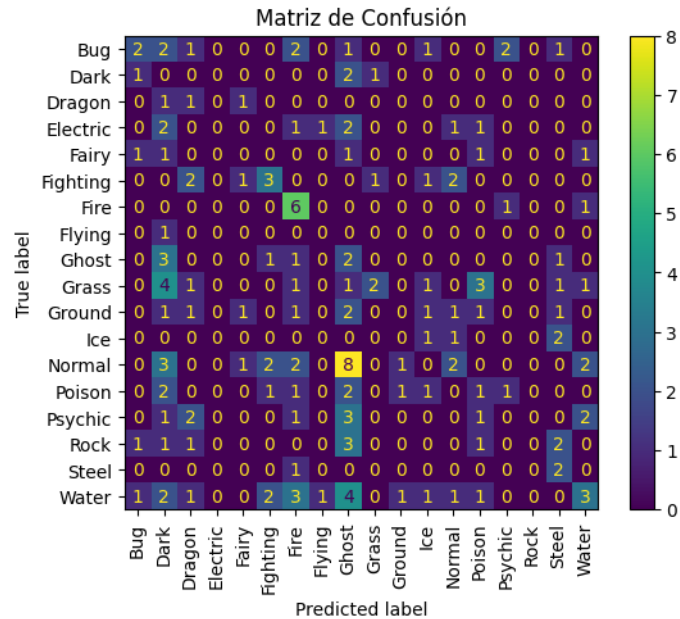
2. Segunda Iteración: Se integró MobileNetV2 como base preentrenada. Esto permitió una extracción más efectiva de características visuales, lo que resultó en una mejora significativa en la precisión.

3. Tercera Iteración: Se optimizaron hiperparámetros como la tasa de aprendizaje, se añadió regularización L2 y se ajustó el tamaño del lote. También se implementaron técnicas de Data Augmentation para mejorar la generalización.

## Resultados

El modelo final logró predecir los tipos de Pokémon con una precisión moderada, aunque hubo dificultades con clases menos representadas. A continuación, se incluyen análisis detallados:





La matriz de confusión reveló que el modelo tiene un buen rendimiento en clases mayoritarias como 'Water' y 'Normal', pero presenta dificultades con clases minoritarias como 'Flying' y 'Fairy'. Esto puede ser atribuido al desbalance de datos, lo que sugiere la necesidad de estrategias adicionales para manejar este problema.



## Conclusión

El proyecto logró implementar un modelo de clasificación de Pokémon basado en imágenes, utilizando MobileNetV2 como base preentrenada. Si bien el modelo mostró resultados prometedores, aún enfrenta desafíos con clases desbalanceadas y con menor representación en el dataset.

Se recomienda explorar técnicas adicionales como el sobremuestreo de clases minoritarias, la recolección de datos adicionales o el uso de modelos específicos para dichas clases. Asimismo, es importante continuar ajustando los hiperparámetros y explorando arquitecturas más avanzadas.

## **Origen de los Datos**

Los datos utilizados en este proyecto han sido extraídos desde el dataset disponible en Kaggle: [Pokémon Images and Types Dataset](#). Este dataset incluye imágenes de Pokémon en formato .png, junto con un archivo pokemon.csv que contiene información detallada de cada Pokémon, como su tipo principal y secundario

## **Bibliografía**

### **1. Documentación de TensorFlow**

- TensorFlow Developers. *TensorFlow API Documentation*. Disponible en: <https://www.tensorflow.org/>.  
Esta fuente se utilizó para implementar las redes neuronales convolucionales y realizar el entrenamiento del modelo.

### **2. MobileNetV2: An Efficient Convolutional Neural Network for Mobile Vision Applications**

- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). *MobileNetV2: Inverted Residuals and Linear Bottlenecks*.  
Disponible en: <https://arxiv.org/abs/1801.04381>.  
Referencia para comprender la arquitectura base utilizada en el modelo.