

Tarea 4

Propuesta de proyecto.

Clasificador Multi-etiqueta de Tipos de Pokémon Basado en imágenes

Sandibel Soares
Alejandro Zambrano
Jesús Cuéllar
Jesús Prieto

Junio 30, 2025

Resumen

Esta propuesta de proyecto trata de la implementación de un clasificador multi-etiqueta que tiene como objetivo el identificar los tipos de un pokémon a través de una imagen del mismo. Para esto es necesario comprender e implementar redes neuronales convolucionales para el análisis de imágenes como datos de entrada y la asignación de hasta dos etiquetas –debido a la naturaleza de los tipos en pokémon–; además de utilizar un dataset actualizado con las 1025 especies de pokémon existentes. Esta es una forma divertida de abordar el problema de multi-etiquetamiento de imágenes, que tiene muchas más aplicaciones.

Introducción

Este proyecto tiene el **objetivo** de implementar un clasificador multi-etiquetas de imágenes, en específico se quiere clasificar imágenes de pokémones y predecir sus tipos.

Un clasificador multi-etiquetas se usa para clasificar datos en casos donde hay dos o más clases/etiquetas, y los datos pueden pertenecer a más de una de ellas. En nuestro caso, se desea clasificar imágenes, tarea que presenta diversas adversidades. Debido a

esto, en este proyecto se decide usar un conjunto de datos con una complejidad reducida: imágenes de pokémones.

Los pokémones son criaturas ficticias del universo *Pokémon*, el cual comenzó como videojuegos RPG. **Una característica esencial de los pokémones es que se asocian a tipos elementales.** Cada tipo tiene resistencias y debilidades a otros tipos, creando un sistema de combate similar a *piedra, papel o tijeras* con muchas más opciones.

En la actualidad, existen 18 tipos de pokémones, por ejemplo: Fuego, Agua, Planta, etc. Cada pokémon tiene uno de estos tipos, o puede ser de tipo dual (tiene dos tipos a la vez). Por otro lado, **su tipo influye en su diseño**, por ejemplo, los pokémones de tipo agua tienden a utilizar tonos azules para establecer una conexión con el agua, o directamente tener características de peces u otros animales acuáticos o semiacuáticos.

Otra influencia en el diseño de un pokémon es la época en la que fueron creados, debido a la adquisición de nuevos artistas o cambios en la idiosincrasia de diseño. Con cada salida de un juego nuevo, inicia una *generación* nueva de pokémones, donde todos los diseños comparten una misma idiosincrasia. Por lo que, otro objetivo de este proyecto es hacer un análisis intergeneracional de los pokémon, es decir, entrenar un modelo con imágenes de una generación, y probarlo con otra generación, para observar si ciertas características entre los tipos se mantuvieron o cambiaron.

Por tanto, el diseño único y simple de los Pokémons, estrechamente ligado a su **tipo**, permite una clasificación **fiable y sin ambigüedad**. Si a esto le sumamos que el universo de Pokémon es un **conjunto finito y manejable** (actualmente con 1025 pokémones diferentes), estas características se presentan como ventajas significativas para el desarrollo y la eficacia del modelo de clasificación que se desea construir.

Como **antecedentes** y fuente de inspiración tenemos al Clasificador Multi-etiqueta de Posters de Películas para predecir sus géneros. Este proyecto tiene múltiples implementaciones, en esta sección presentamos un artículo sencillo y explicativo del mismo en la siguiente página: [link](#) (última consulta: 30/06/2025).

Fuera del ejemplo que elegimos, el campo de clasificadores multi-etiqueta es amplio, incluso dentro de la subcategoría de clasificación de imágenes, por lo que contamos con múltiples antecedentes, ubicados en la Bibliografía, bajo los títulos “**Proyectos similares**” y “**Clasificadores multi-etiqueta y clasificadores de imágenes**”.

En el caso específico de un clasificador de tipos de pokémones, tenemos este proyecto de GitHub como ejemplo: [link](#) (última consulta: 30/06/2025), el cual hace uso de una red neuronal convolucional (CNN).

A través de este proyecto, buscamos **reforzar y adquirir nuevos conceptos en el área de Aprendizaje de Máquinas**. Específicamente, nos enfocaremos en profundizar sobre distintas clases de redes neuronales, como las convolucionales, y en la implementación de sistemas de clasificación multi-etiqueta, esta última técnica es de gran interés, ya que se aproxima mejor a los problemas del mundo real al permitir que una imagen tenga más de una etiqueta asociada. Adicionalmente, trabajaremos con el

procesamiento de imágenes como entrada para el modelo. Todo este aprendizaje práctico es de gran valor, sin embargo, nuestra mayor motivación en este proyecto es lo **divertido** y **creativo** que resulta ser.

Objetivo

El objetivo del proyecto es implementar un clasificador multi-etiqueta de imágenes con una red neuronal convolucional, con el objeto de predecir los tipos de un pokémon dada su imagen.

Diseño previo de la solución

Debido a la naturaleza del proyecto, se tiene que se usará técnicas de **aprendizaje supervisado**.

Se desea **desarrollar** una **red neuronal convolucional** para la clasificación de imágenes cuya capa de salida corresponda a una capaz de asignar múltiples etiquetas (a lo sumo dos) para cada pokémon. Sin embargo, como apoyo, y para reforzar el aprendizaje, también se desea implementar el modelo por medio librerías siguiendo a grandes rasgos el proyecto anteriormente mencionado de las [películas](#). Entre las librerías que se utilizarían se tiene:

Keras (Usada para construir y entrenar redes neuronales), **Numpy** (utilizada para trabajar con arreglos y operaciones matemáticas de forma eficiente. **Pandas** (Permite trabajar con estructuras de datos tipo tabla facilitando su limpieza y manipulación), **Matplotlib** (gráficas y visualizar datos) y **Scikit-learn** (División de los datos en conjuntos de entrenamiento y prueba)

Experiencia con la que se entrenará la máquina a desarrollar

Para la búsqueda de conjuntos de datos destinados a usarse como experiencia para el entrenamiento se utilizó la página Kaggle. Entre los posibles conjuntos de datos a usar se escogió ***“Dataset of 32000 Pokemon Images & CSV, JSON”*** [0]

Este conjunto de datos incluye dos directorios con aproximadamente 32000 imágenes de Pokémon, junto un archivo CSV y un archivo JSON. El archivo CSV contiene información sobre cada Pokémon, como su nombre, tipo, especie, altura, peso, habilidades, rendimiento de EV, tasa de captura, amistad base, experiencia base, tasa de crecimiento, grupos de huevos, género, ciclos de huevos y estadísticas base.

Del conjunto de datos nos interesa:

- La carpeta de imágenes llamada ***“Pokemon Images DB”*** en la cual hay 1187 imágenes de pokémones (distintos), cada una acompañada de una imagen original y una imagen sin fondo.
- El archivo CSV ***“pokemonDB_dataset.csv”*** del cual nos interesan las columnas ***“nombre”*** (Para poder acceder a la imagen correspondiente al pokémon) y la columna ***“tipo”*** la cual contiene la lista de tipos a las que pertenece el pokémon (Estos tipos serán las etiquetas objetivo)

Adicionalmente, se añadirá por cuenta propia una columna adicional llamada “**generación**”, con el objetivo de realizar las pruebas intergeneracionales. Para esto se usará el número de Pokédex Nacional provisto por la fuente del dataset elegido: [PokémonDB](#).

Bibliografía

DataSet

[0] *Dataset of 32000 Pokemon Images & CSV, JSON*. (2024, 28 marzo). Kaggle.
<https://www.kaggle.com/datasets/divyanshusingh369/complete-pokemon-library-32k-images-and-csv/data>

[1] *Pokémon Pokédex: list of Pokémon with stats*. (s. f.). Pokémon Database.
<https://pokedex.net/pokedex/all>

Proyectos similares

[1] Sharma, P. (2025, 13 enero). *Build your First Multi-Label Image Classification Model in Python*. Analytics Vidhya.
<https://www.analyticsvidhya.com/blog/2019/04/build-first-multi-label-image-classification-model-python/>

[2] Akhiljanardhanan. (2021, 15 diciembre). Multi-Label Image Classification using CNN (python). *Medium*.
<https://medium.com/@akhiljanardhanan525/multi-label-image-classification-using-cnn-python-9c5dbddf7dc7>

[3] Movie Genre Classification based on Poster Images with Deep Neural Networks
<https://mmcv.csie.ncku.edu.tw/~wtchu/papers/2017MUSA-chu.pdf>

[4] Multi-Label Classification of Satellite Photos of the Amazon Rainforest
<https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-to-classify-satellite-photos-of-the-amazon-rainforest/>

[5] Retinskiy, D., & Retinskiy, D. (2021, 5 mayo). *Multi-Label Image Classification with PyTorch* <https://learnopencv.com/multi-label-image-classification-with-pytorch/>

[6] Bebnev, V., & Bebnev, V. (2021, 4 mayo). *Multi-Label Image Classification with PyTorch: Image Tagging*
<https://learnopencv.com/multi-label-image-classification-with-pytorch-image-tagging/>

[7] Joshi, P. (2024, 16 octubre). *Predicting Movie Genres using NLP – An Awesome Introduction to Multi-Label Classification*. Analytics Vidhya.

<https://www.analyticsvidhya.com/blog/2019/04/predicting-movie-genres-nlp-multi-label-classification/>

[8] Davideiacobs. (s. f.). *GitHub - davideiacobs/-Movie-Genres-Classification-from-their-Poster-Image-using-CNNs*. GitHub. <https://github.com/davideiacobs/-Movie-Genres-Classification-from-their-Poster-Image-using-CNNs>

[9] Rshnn. (s. f.). *GitHub - rshnn/pokemon-types: deep convolutional neural network for Pokemon type classification using image data*. GitHub. <https://github.com/rshnn/pokemon-types?tab=readme-ov-file>

[10] *Pokemon types image detection · Models · Dataloop*. (s. f.). https://dataloop.ai/library/model/dima806_pokemon_types_image_detection/

Clasificadores multi-etiquetas y clasificadores de imágenes

[9] Learning multi-label scene classification, Matthew R. Boutella , Jiebo Luob <https://research.csc.ncsu.edu/picture/publications/papers/pr04.pdf>

[10] GeeksforGeeks. (2024, 6 enero). *Multiclass Classification vs Multilabel Classification*. GeeksforGeeks. <https://www.geeksforgeeks.org/multiclass-classification-vs-multi-label-classification/>

CNN

[11] Singh, A. (2020, 8 mayo). *Demystifying the Mathematics Behind Convolutional Neural Networks (CNNs)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/?utm_source=blog&utm_source=learn-image-classification-cnn-convolutional-neural-networks-5-datasets

[12] Rizvi, M. S. Z. (2025, 1 mayo). *Image classification using CNN*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2020/02/learn-image-classification-cnn-convolutional-neural-networks-3-datasets/>

[13] Singh, A. (2020b, mayo 8). *Demystifying the Mathematics Behind Convolutional Neural Networks (CNNs)*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/02/mathematics-behind-convolutional-neural-network/?utm_source=blog&utm_source=learn-image-classification-cnn-convolutional-neural-networks-5-datasets

[14] Tamanna. (2023, 24 abril). Exploring Convolutional Neural Networks: Architecture, Steps, Use Cases, and Pros and Cons. *Medium*. <https://medium.com/@tam.tamanna18/exploring-convolutional-neural-networks-architecture-steps-use-cases-and-pros-and-cons-b0d3b7d46c71>

[15] Opeyemi, A. T. (2021, 24 diciembre). Multilabel Classification with CNN - AKINTIBU TOSIN OPEYEMI - Medium. *Medium*.
<https://aoakintibu.medium.com/multilabel-classification-with-cnn-278702d98c5b>

Librerías

[17] TensorFlow, “TensorFlow: An end-to-end open source machine learning platform.” [En línea]. Disponible en: <https://www.tensorflow.org/>

[18] Keras, “Keras: Deep Learning for humans.” [En línea]. Disponible en: <https://keras.io/>

[19] PyTorch, “PyTorch: An open source machine learning framework.” [En línea]. Disponible en: <https://pytorch.org/>

[20] NumPy, “NumPy: The fundamental package for scientific computing with Python.” [En línea]. Disponible en: <https://numpy.org/>

[21] Pandas, “Pandas: Python Data Analysis Library.” [En línea]. Disponible en: <https://pandas.pydata.org/>

[22] Matplotlib, “Matplotlib: Visualization with Python.” [En línea]. Disponible en: <https://matplotlib.org/>

[23] scikit-learn, “scikit-learn: Machine Learning in Python.” [En línea]. Disponible en: <https://scikit-learn.org/>