

Clasificación de imágenes satelitales (VIT)

Carlos Christopher Gaubert Quijada

Universidad del Bío-Bío, Avda. Collao 1202 Casilla 5-C, Concepción, Bío Bío

<https://github.com/CarlosGaubertQ/trabajosemestralML/tree/main>

Abstract

El presente informe aborda el problema crítico de la deforestación y los incendios forestales, fenómenos de gran relevancia a nivel global debido a sus devastadoras consecuencias para el medio ambiente y la biodiversidad. Con el propósito de abordar esta problemática, se propuso la implementación de tecnologías basadas en el aprendizaje automático supervisado y la arquitectura "Vision Transformer" (ViT) para identificar y monitorear zonas específicas donde ocurren estos eventos. Los resultados de la investigación proporcionaron una comprensión profunda del funcionamiento. Esto se convirtió en la base fundamental para guiar la implementación del sistema, asegurando un esquema exitoso para su aplicación. En conclusión, este proyecto representa un valioso aporte en la lucha contra la deforestación y los incendios forestales mediante el uso de tecnologías avanzadas que permiten una identificación y monitoreo más eficiente de estas problemáticas ambientales. La combinación de aprendizaje automático supervisado y la arquitectura Vision Transformer presenta un enfoque innovador y prometedor para abordar este desafío global, sentando las bases para futuras investigaciones y acciones encaminadas a preservar nuestros ecosistemas y recursos naturales.

Keywords: Deforestación, Incendios forestales, Aprendizaje automático supervisado, Vision Transformers, Teledetección, Imágenes satelitales, Segmentación, Software, Python, Django.

1. Motivación

Este proyecto fue realizado con la finalidad de poder comprender los principios de las redes neuronales las cuales han tenido un alto impacto en los últimos años en relación al mundo laboral tanto como para informáticos, como de otras áreas, dicho esto se planea comprender a fondo el funcionamiento de la librería Pytorch la cual es fundamental para poder implementar esta nueva herramienta de trabajo.

2. Software

Para tener una breve explicación del proyecto lo que se plantea es entrenar 3 modelos de redes neuronales las cuales son:

- CNN multi clase
- ViT multi clase
- ViT Binario

2.1. Entrenamiento Redes Neuronales

En el desarrollo de redes neuronales para la presente investigación, se optó por la utilización de la biblioteca PyTorch. Esta elección se fundamenta en la extensa gama de herramientas que proporciona PyTorch, permitiendo la creación meticulosa de modelos. A diferencia de otras librerías de Machine Learning y Deep Learning, que ofrecen funciones y aplicaciones listas para su implementación rápida, PyTorch se destaca por ser de bajo nivel.

En este contexto, se observa que PyTorch proporciona mayor flexibilidad y control en la creación de modelos, permitiendo un enfoque más exhaustivo. Aunque otras librerías ofrecen soluciones predefinidas para implementaciones rápidas, PyTorch brinda a los investigadores la capacidad de personalizar cada aspecto del modelo de manera más detallada. Este enfoque de bajo nivel proporciona a los desarrolladores una mayor comprensión y manipulación de los elementos fundamentales de las redes neuronales, lo cual resulta especialmente beneficioso en investigaciones que requieren un nivel más profundo de ajuste y optimización de modelos.

2.2. Tratamiento del conjunto de datos

Para la utilización del conjunto de datos en esta investigación, se ha optado por emplear EUROSAT, un dataset especialmente diseñado para llevar a cabo la clasificación de imágenes satelitales. Este conjunto de datos abarca una amplia diversidad de entornos, entre los cuales se incluyen categorías como AnnualCrop, Forest, HerbaceousVegetation, Highway, Industrial, Pasture, PermanentCrop, Residential, River, y SeaLake.

2.3. Separación del conjunto de datos

En la fase de preprocesamiento de datos, se tomó la decisión de estructurar el conjunto de datos EUROSAT en tres formatos distintos, cada uno diseñado para abordar objetivos específicos de entrenamiento y evaluación del modelo. El primer formato consiste en un conjunto de 10 clases, donde se asignó el 80% de los datos para el entrenamiento y el 20% para las pruebas. Esta configuración busca lograr un equilibrio entre la cantidad de datos utilizados para el aprendizaje y la evaluación del rendimiento del modelo en diversas clases. En el segundo formato, se optó por trabajar nuevamente con las 10 clases, asignando un 90% de los datos para el entrenamiento y reservando el 20% para las pruebas. Este enfoque busca evaluar el impacto de un conjunto de entrenamiento más extenso en la capacidad del modelo para generalizar y clasificar eficientemente las distintas categorías. El tercer formato se diseñó específicamente para abordar la clasificación binaria entre bosques y no bosques. En este caso, se seleccionaron dos clases: la clase de bosques y una clase compuesta por la amalgama de las demás categorías. El conjunto se dividió con un 80% de los datos destinados al entrenamiento y un 20% para las pruebas. Este enfoque binario busca crear modelos especializados en la identificación de áreas forestales, con el desafío adicional de distinguir entre la presencia y ausencia de bosques en paisajes diversos. Estas configuraciones permitirán evaluar el rendimiento del modelo en diferentes escenarios, brindando una comprensión más completa de su capacidad para clasificar imágenes satelitales en función de la estructura del conjunto de datos y los objetivos específicos de cada formato.

References

- [1] P. Helber, B. Bischke, A. Dengel and D. Borth, "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 12, no. 7, pp. 2217-2226, July 2019, doi: 10.1109/JSTARS.2019.2918242.

Illustrative Examples

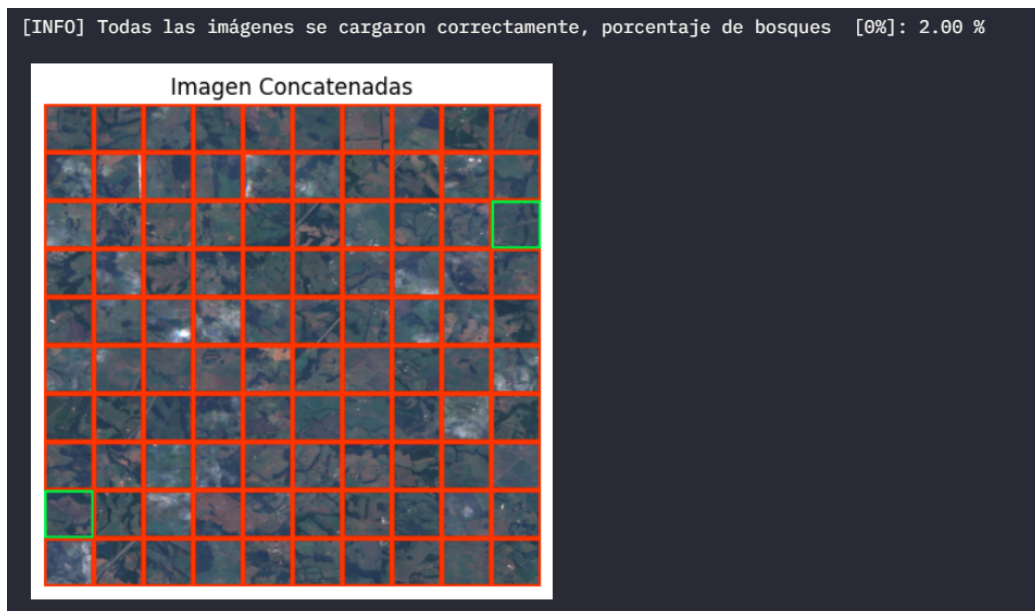


Figure 1: Ejemplo uso de modelo ViT multiclase 1



Figure 2: Ejemplo uso de modelo ViT multiclase 2

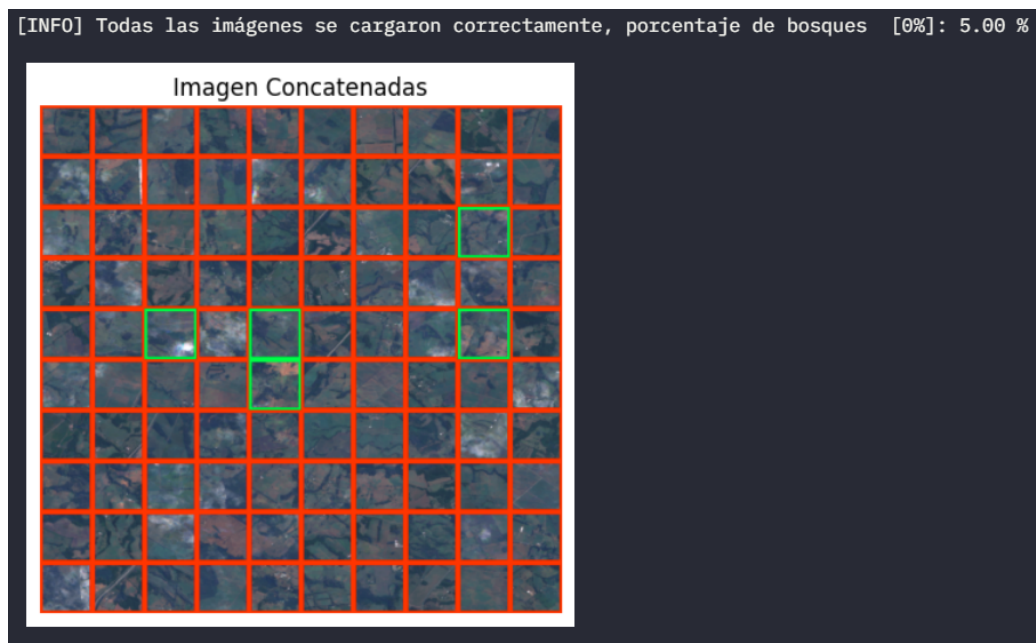


Figure 3: Ejemplo uso de modelo ViT Binario 1



Figure 4: Ejemplo uso de modelo ViT Binario 2



Figure 5: Ejemplo uso de modelo CNN multiclase 1

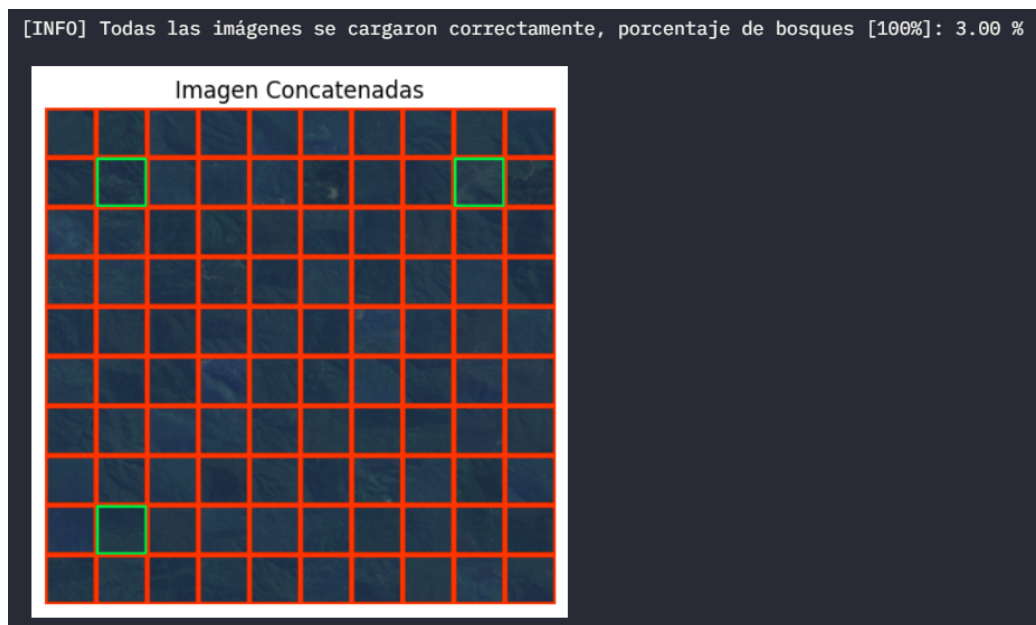


Figure 6: Ejemplo uso de modelo CNN multiclase 2

3. Conclusión

El proyecto reveló una marcada disparidad entre el empleo de redes neuronales convolucionales (CNN) y transformers. Se observó que la utilización de transformers, en comparación con las CNN, conlleva una mejora significativa en la eficacia del reconocimiento de patrones en imágenes reales. Esta mejora radica en la capacidad de los transformers para preservar las características fundamentales de las imágenes, a diferencia de las CNN, que tienden a destruir información relevante durante el proceso de convolución.

A pesar de que quedaron algunas pruebas pendientes, los resultados obtenidos hasta el momento reflejan un desempeño positivo, alineado con los objetivos planteados al inicio del semestre. La implementación realizada permitió alcanzar buenos resultados y proporcionó una visión clara de las ventajas que ofrece el uso de transformers en la tarea de reconocimiento de patrones en imágenes reales. Este hallazgo respalda la idea de que la elección de la arquitectura del modelo desempeña un papel crucial en el éxito de los proyectos de visión por computadora, destacando la relevancia de considerar enfoques innovadores, como los transformers, para optimizar el rendimiento del sistema.

Required Metadata

Current code version

Nr.	Code metadata description	Please fill in this column
C1	Current code version	v1
C2	Permanent link to code/repository used for this code version	https://github.com/CarlosGaubertQ/trabajosemestralML
C3	Permanent link to Reproducible Capsule	
C4	Legal Code License	
C5	Code versioning system used	Git
C6	Software code languages, tools, and services used	Python, Django, Opencv, Google Earth Engine
C7	Compilation requirements, operating environments & dependencies	
C8	If available Link to developer documentation/manual	https://github.com/CarlosGaubertQ/trabajosemestralML
C9	Support email for questions	

Table 1: Code metadata (mandatory)

Current executable software version

Nr.	(Executable) software meta-data description	Please fill in this column
S1	Current software version	1.1
S2	Permanent link to executables of this version	https://github.com/CarlosGaubertQ/trabajosemestralML
S3	Permanent link to Reproducible Capsule	
S4	Legal Software License	
S5	Computing platforms/Operating Systems	Windows, Linux and Mac OS
S6	Installation requirements & dependencies	
S7	If available, link to user manual - if formally published include a reference to the publication in the reference list	https://github.com/CarlosGaubertQ/trabajosemestralML
S8	Support email for questions	

Table 2: Software metadata (optional)